

AD-A239 857

DTIC

2



NPS-ME-91-061

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



**A GENERAL-PURPOSE FINITE -DIFFERENCE  
CODE FOR SOLVING STEADY-STATE  
THREE-DIMENSIONAL FLUID-FLOW AND  
HEAT TRANSFER PROBLEMS**

S.B. Sathe, Y. Joshi, and M.D. Kelleher

July 1991

Final Report for Period Jan 1990-Dec 1990

Approved for Public Release; Distribution Unlimited

Prepared for: Naval Weapons Support Center  
Crane, Indiana 47522

91-09104

**NAVAL POSTGRADUATE SCHOOL**  
Monterey, California 93943

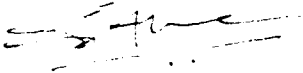
Rear Admiral R.W. West, Jr.  
Superintendent

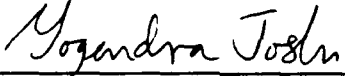
H. Shull  
Provost

This report was prepared in conjunction with research sponsored in part by the Naval Weapons Support Center, Crane, Indiana.

Reproduction of all or part of this report is authorized.

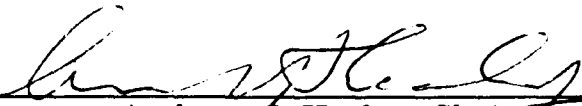
This report was prepared by:

  
Sanjeev B. Sathe, Adjunct Professor

  
Yogendra Joshi, Associate Professor

  
Matthew D. Kelleher, Professor

Reviewed by:

  
Anthony J. Healey, Chairman,  
Department of Mechanical Engineering

Released by:

  
Paul J. Marto  
Dean of Research

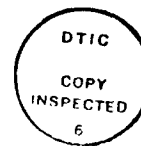
Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION <b>Unclassified</b>			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION AVAILABILITY OF REPORT		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			Approved for public release; distribution unlimited.		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) <b>NPS-ME-91-001</b>			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION <b>Naval Postgraduate School</b>		6b OFFICE SYMBOL (If applicable) <b>ME</b>		7a. NAME OF MONITORING ORGANIZATION <b>Naval Postgraduate School</b>	
6c. Address (City, State, and ZIP Code) <b>Monterey, CA 93943-5000</b>				7b. Address (City, State, and ZIP Code) <b>Monterey, CA 93943-5000</b>	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION <b>NWSC, Crane</b>		8b OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) <b>Crane, IN 47522</b>		10. SOURCE OF FUNDING NUMBERS			
		PROGRAM ELEMENT No.		PROJECT No.	TASK No.
				WORK UNIT ACCESSION No.	
11. TITLE (Include Security Classification) <b>A General -Purpose Finite-Difference Code For Solving Steady-State Three-Dimensional Fluid Flow and Heat Transfer Problems</b>					
12. PERSONAL AUTHOR(S) <b>HAUKENES, LARRY O.</b>					
13a. TYPE OF REPORT <b>Technical Report</b>		13b. TIME COVERED From: <b>Jan. 1990</b> To: <b>Dec. 1990</b>		14. DATE OF REPORT (Year, Month, Day) <b>1991, June, 26</b>	
				15. PAGE COUNT <b>54</b>	
16. SUPPLEMENTARY NOTATION <b>The views expressed in this report are those of the authors and do not reflect the official policy or position of the Department of Defense or the U. S. Government.</b>					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Computational modeling, numerical heat transfer, convective cooling		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>A general purpose FORTRAN code is described for the computations of three-dimensional steady heat transfer and fluid flow in rectangular geometries. The code is designed to handle combined conduction-convection problems within domains comprising of both solid and fluid regions. This is accomplished through the use of the harmonic mean formulation for evaluating interface diffusivities. A control volume formulation is used to discretize the governing equations for the dependent variables in the primitive form. The velocity-pressure coupling is handled by the SIMPLER algorithm. The iterative solution scheme employs the tri-diagonal matrix algorithm of Thomas. A sample example of three-dimensional natural convection in a rectangular enclosure is used to validate the code against existing numerical solutions.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION <b>Unclassified</b>		
22a. NAME OF RESPONSIBLE INDIVIDUAL <b>Yogendra Joshi</b>			22b TELEPHONE (Include Area code) <b>(408) 646-3400</b>		22c. OFFICE SYMBOL <b>ME/Ji</b>

# CONTENTS

Section		Page
1	Scope of the program.....	1
2	Program structure.....	1
3	Program usage.....	3
4	Flow chart.....	11
<b>Appendix</b>		
A	Description of FORTRAN variables, computational domain and the control volume arrangement.....	13
B	Program source code.....	18
C	Example problem: natural convection in a rectangular box.....	46
<b>References .....</b>		<b>54</b>



Acquisition For	
DTIC	<input checked="" type="checkbox"/>
DTIC Doc	<input type="checkbox"/>
Unpublished	<input type="checkbox"/>
Subscription	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

## 1. Scope of the Program

The program is a general purpose finite-difference FORTRAN-77 code designed to solve steady-state three-dimensional heat transfer and fluid flow problems in rectangular coordinates. The code is designed to effectively handle conjugate domain problems, arising when both solid and fluid are present in the computational region. Pure heat conduction problems or only fluid flow problems can also be solved.

A control-volume formulation is used to discretize the governing equations expressed in a canonical form. The control volumes for the velocity components are staggered with respect to those for the temperature and pressure. Harmonic mean formulation for the interface diffusivities is used to effectively handle sharp discontinuities in property values in the computation domain, without having to resort to very fine mesh in the region of discontinuities. The velocity-pressure coupling is handled by the SIMPLER algorithm of Patankar [1]. The solution to the steady state problem is achieved by providing an initial guess of all the dependent variables and proceeding through an iterative scheme using the Thomas algorithm (also known as the Tri-Diagonal Matrix Algorithm). Iterations are to be stopped when values of variables in successive iterations do not change more than a specified convergence criterion.

## 2. Program Structure

The code is highly modular consisting of a short main program and various subroutines. The problem independent portion consists of the main program and subroutines **PROFIL**, **TDMA**, **FORM** and **OPTION**. The

problem dependent part, which the user is primarily concerned with, consists of a **BLOCK DATA** subprogram and a subroutine **USE**.

A brief description of the various subprograms/subroutines follows:

- PROFIL** The profile assumptions in between node-points for the dependent variables are incorporated here. Presently, it contains the power law profiles [1]. Other profile schemes such as the hybrid scheme or the central differencing scheme may be incorporated by the user.
- TDMA** The line-by-line TDMA (**Tri-Diagonal Matrix Algorithm**) is used for solving the equations. The subroutine solves for a canonical equation described later (see Eq. 1).
- FORM** Evaluates the geometric parameters such as lengths, areas etc. in **ENTRY GEOMET** and the coefficients for the various equations in **ENTRY COEFF** which are then used to solve for various equations in accordance with the **SIMPLER** algorithm [1].
- OPTION** The user may access two optional utilities (entries), **UMESH** and **PRINT**. **UMESH** is to be accessed if a uniform grid is required, while **PRINT** is called if dependent variables are to be printed out over the entire domain. Calling **PRINT** causes a plane by plane field printout, for different planes perpendicular to the z axis.
- BLOCK DATA** Contains block **COMMONS** that contain default values of various quantities described subsequently. The default values can be changed by the user.

**USE** Contains problem dependent information to be specified by the user. The different entries in the subroutine **USE** are **MESH, BEGIN, VARRHO, BNDRY, PRTOUT** and **DIFFUS**. The usage of these entries is described later.

### 3. Program Usage

It is required at first to cast the equations to be solved in the following canonical form:

$$\frac{\partial}{\partial x}(\rho u \phi) + \frac{\partial}{\partial y}(\rho v \phi) + \frac{\partial}{\partial z}(\rho w \phi) = \frac{\partial}{\partial x}(\Gamma \frac{\partial \phi}{\partial x}) + \frac{\partial}{\partial y}(\Gamma \frac{\partial \phi}{\partial y}) + \frac{\partial}{\partial z}(\Gamma \frac{\partial \phi}{\partial z}) + S \quad (1)$$

where  $\rho$  is the fluid density,  $u$ ,  $v$  and  $w$  are the velocity components in the  $x$ ,  $y$  and  $z$  directions respectively,  $\phi$  is the dependent variable to be solved for,  $\Gamma$  is the appropriate diffusivity and  $S$  is the source term.

For efficient execution of the algorithm, and for better convergence prospects, it is necessary that  $S$  be represented in the form

$$S = S_c + S_p \phi \quad (2)$$

and it is required that  $S_p \leq 0$ . For an appropriate linearization of the source term  $S$ , when it does not have a readily obtainable form of Eq. (2), the user is referred to Patankar [1], pp. 48-49.

It is not required to cast the continuity equation into the canonical form, since this is incorporated in the program internally. For example, when solving a combined incompressible fluid-flow, heat transfer problem, the user needs to cast the three momentum equations and one energy equation in the canonical form of Eq. (1). Note that the pressure gradient terms encountered in the momentum equations are incorporated internally and the user is cautioned not to incorporate them in the source term S.

As mentioned earlier, the user needs to be concerned only with the **BLOCK DATA** and subroutine **USE**. A detailed description of these segments now follows.

### **BLOCK DATA**

The array  $F(I,J,K,NF)$  contains different variables or properties. The default equivalence of the array  $F$  is given below:

<u>NF</u>	<u>Quantity Stored in <math>F(I,J,K,NF)</math></u>	<u>FORTRAN Array in <b>USE</b></u>
1	velocity in x direction, u	$U(I,J,K)$
2	velocity in y direction, v	$V(I,J,K)$
3	velocity in z direction, w	$W(I,J,K)$
4	pressure correction	$PC(I,J,K)$
5	temperature	$T(I,J,K)$
6	pressure	$P(I,J,K)$



The **BLOCK DATA** contains several data that may be reinitialized by the user. Each of these is explained below:

IPREF, JPREF, KPREF -

Values of the indices (I,J,K) where the pressure is set to zero. Note that the program evaluates only pressure differences and not the actual pressures.

NTIMES -

The value of NTIMES(NF) determines the number of sweeps of the TDMA solution over the entire domain during one iteration. More than one sweep per iteration may be required for highly nonlinear equations to assure convergence. Clearly, a larger value of NTIMES(NF) will result in a longer computational time.

LAST -

Defines the total number of iterations to be performed.

LPRINT -

The default value of this logical FORTRAN variable is FALSE. Setting LPRINT(NF) to be true will cause a field printout of the particular variable associated with NF when a call to the entry PRINT is invoked through the statement CALL PRINT.

LSOLVE -

The default value of this logical FORTRAN variable is FALSE. Setting LSOLVE(NF) to be true causes the solution of the particular variable associated with NF. In a pure conduction problem, for example, only LSOLVE(5) must be set

true. In a pure fluid flow problem (no heat transfer) LSOLVE(NF), NF=1,2,3,4, and 6 must be declared true. In a heat transfer and fluid flow problem, LSOLVE(NF), NF=1 to 6 should be set true.

**RELAX -** The value for RELAX(NF) is the underrelaxation factor for the solution of the variable associated with NF. RELAX(NF)>1 causes overrelaxation, while RELAX(NF)<1 causes underrelaxation. Underrelaxation may be necessary in many problems when convergence is difficult to obtain, and will increase the computational time.

**RHOCON -** Value of  $\rho$  in Eq.(1) is specified here.

**TITLE(NF) -** This describes the title to be given to each field printout.

Subroutine **USE** contains various segments (entries) called by the main program. The purpose of these and their usage is described here. See **Appendix A** for meaning of the various symbols.

**MESH:** The user supplies a mesh in this entry. The computational domain must be divided into rectangular control volumes. The internal grid points are placed at the geometric centers of the control volumes. The boundary grid points are placed on the geometric centers on the control volumes faces that form part of the boundary (See Figs. A-1 and A-2 in Appendix A).

Two options are available-

1. Uniform grid - The user supplies values of L1, M1 and N1; XL, YL and ZL (see Appendix A); and then invokes CALL UMESH to form a uniform mesh, i.e. all the control volumes in a particular direction have the same length:

```
ENTRY MESH
L1=10
M1=10
N1=20
XL=2.
YL=3.
ZL=1.
CALL UMESH
RETURN
```

This implies a computational domain that is 2, 3 and 1 units long in the x, y and z directions respectively. All the control volumes thus have dimensions of 2/10, 3/10 and 1/20 units in the x, y and z directions respectively.

2. Non-uniform grid - The user supplies values of (XU(I), I=1,L1), (YV(J), J=1,M1), ZW(K), K=1,N1); XL, YL, ZL; L1, M1 and N1. The code internally determines the locations of the boundary and internal grid points (stored in X(I),Y(J),Z(K)) which are placed at the geometric centers of the control volumes. For example,  $X(I)=0.5*[XU(I)+XU(I+1)]$  etc. Boundary points are grid points that are located on planes  $x=0$  or  $x=XL$  or  $y=0$  or  $y=YL$  or  $z=0$  or  $z=ZL$ .

**BEGIN:** The user specifies/calculates quantities that are required prior to the start of iterations such as parameter values, initial guess to the solution and boundary conditions.

Note: Entries **MESH** and **BEGIN** are called by the main program only once before the iterations begin.

**VARRHO:** Functional dependency of the fluid density can be accounted for in this entry for variable density situations.

**BNDRY:** The boundary values of the variables are updated every iteration in accordance with the boundary conditions.

Note: Boundary conditions that involve specification of the dependent variable value can be incorporated in **BEGIN** and it is not necessary to include these in **BNDRY**. This is so because the program solves for the dependent variables only at the internal grid points and not at the boundary points.

Examples of different boundary conditions:

```
ENTRY BNDRY
DO 1 I=1,L1
DO 1 J=1,M1
C   Insulated boundary
C   At  $z=ZL$ ,  $dT/dz=0$ 
T(I,J,N1)=T(I,J,N1-1)
C   Convective condition
C   At  $z=0$ ,  $h(T_f-T)=-kdT/dz$ 
C    $H=h$  (convection coefficient),  $TF=T_f$  (fluid temperature)
C    $TK=k$  (thermal conductivity)
DELZ=Z(2)-Z(1)
T(I,J,1)=(H*DELZ*TF+TK*T(I,J,2))/(TK+H*DELZ)
C   Specified heat flux condition
C   at  $z=ZL$ ,  $-kdT/dz=-q''$ ,  $q''>0$ 
C    $QPRIME=q''$  (heat flux)
DELZ=Z(N1)-Z(N1-1)
T(I,J,N1)=T(I,J,N1-1)+QPRIME*DELZ/TK
1  CONTINUE
```

## RETURN

**PRTOUT:** It may be necessary to monitor values of different variables or other quantities of interest as each iteration proceeds. This can be done in ENTRY PRTOUT. At the end of the iterations, CALL PRINT invokes a field printout of the variables for which LPRINT has been set true.

**DIFFUS:** The user specifies  $\Gamma$ ,  $S_c$  and  $S_p$  (Eqs. 1,2) in the variables GAM(I,J,K), CON(I,J,K) and AP(I,J,K), respectively. The diffusivity  $\Gamma$  is specified at all points, while  $S_c$  and  $S_p$  are specified at the internal points (i.e. all points excluding the boundary points). For example:

### ENTRY DIFFUS

```
C      Specify NF since DIFFUS is called by the main program
C      more than once every iteration.
      DO 2 I=1,L1
      DO 2 J=1,M1
      DO 2 K=1,N1
C      VISC is the viscosity

      IF((NF.EQ.1).OR.(NF.EQ.2).OR.(NF.EQ.3))GAM(I,J,K)=
1VISC
C      TK is the thermal conductivity, CP is the specific heat
C      capacity.
      IF(NF.EQ.5)GAM(I,J,K)=TK/CP
2      CONTINUE
      ENDIF
C      In the event of uniform volumetric heat generation in
C      the domain (Q W/m3).
      DO 3 I=2,L1-1
      DO 3 J=2,M1-1
      DO 3 K=2,N1-1
      IF(NF.EQ.5)CON(I,J,K)=Q/CP
C      If AP(I,J,K) is not specified, it is zero by FORTRAN
C      default.
```

```

3      CONTINUE

      .
      .
C      SPECIAL TREATMENT OF FLUX TYPE BOUNDARY
C      CONDITIONS (TO IMPROVE CONVERGENCE)
C      SET BOUNDARY VALUE OF DIFFUSIVITY TO ZERO.
C      CONVERT SURFACE HEAT FLUX TO AN
C      EQUIVALENT HEAT SOURCE IN THE CONTROL
C      VOLUME ADJACENT TO THE BOUNDARY.
      DO 4 I=1,L1
      DO 4 J=1,M1
C      Insulated boundary
C      At  $z=ZL$ ,  $dT/dz=0$ ,
      IF(NF.EQ.5)GAM(I,J,N1)=0.0
C      Convective condition
C      At  $z=0$ ,  $h(T_f-T)=-kdT/dz$ 
C       $S$ =Equivalent heat source/volume=surface heat flux*
C      surface area/volume (source term for next to boundary
C      node).
C      Surface convective flux= $h(T_f-T)=(T_f-T_i)/[1/h+(z_i-z_b)/k]$ 
C      where  $T_i$  is the temperature at the node adjacent
C      to the boundary,  $z_i$  is the  $z$  location of the node,  $z_b$  is the
C       $z$  location of the boundary. The factor in square brackets
C      is the thermal resistance ( $R_{th}$ ) due to convection and
C      conduction.
C      Surface area=XCV(I)*YCV(J).
C      Volume of control volume=XCV(I)*YCV(J)*ZCV(2).
C      Surface area/volume=1./ZCV(2)
C       $T_i = T(I,J,2)$ ,  $z_i-z_b=Z(2)-Z(1)$ ,  $T=T(I,J,1)$ .
C      Source  $S$  may be split into  $S_c$  and  $S_p$ 
C       $S_c=(T_f/R_{th})/ZCV(2)$ ,  $S_p=-1/(R_{th}*ZCV(2))$ 
      RTH=(1./H+(Z(2)-Z(1))/TK)
      IF(NF.EQ.5)THEN
      GAM(I,J,1)=0.0
C      Add  $S_c$  and  $S_p$  for the control volume adjacent to the
C      boundary.
      CON(I,J,2)=CON(I,J,2)+TF/(RTH*ZCV(2))
      AP(I,J,2)=AP(I,J,2)-1./(RTH*ZCV(2))
      ENDIF
C      Specified heat flux condition
C      At  $z=ZL$ ,  $-kdT/dz=-q''$ ,  $q''>0$ 
C      QPRIME= $q''$  (heat flux)
      IF(NF.EQ.5)THEN
      GAM(I,J,N1)=0.0
      CON(I,J,N1-1)=CON(I,J,N1-1)+QPRIME/ZCV(N1-1)
      ENDIF
4      CONTINUE
      RETURN

```

**Conjugate Domains**      Frequently it is required to calculate heat transfer in domains that contain both fluid and solid. The solid then can be easily modeled by setting the fluid viscosity to a very large value (eg.  $10^{20}$ ) in the solid region of the computational domain. The velocities in that region then will be computed by the code to be zero. Appropriate solid properties such as thermal conductivity can be prescribed in the solid region. Care must be taken that the control volume faces coincide with the actual physical discontinuities in the domain.

#### **4. Program Flow-chart**

The flow-chart for the program execution is shown in Fig. 1. The usage and the purpose of the various subroutines and subprograms has already been explained in the preceding sections. A single iteration consists of execution of the SIMPLER algorithm to evaluate the velocity and pressure field in the domain and then the solution of any other dependent variables to be solved for, based on the calculated velocities.

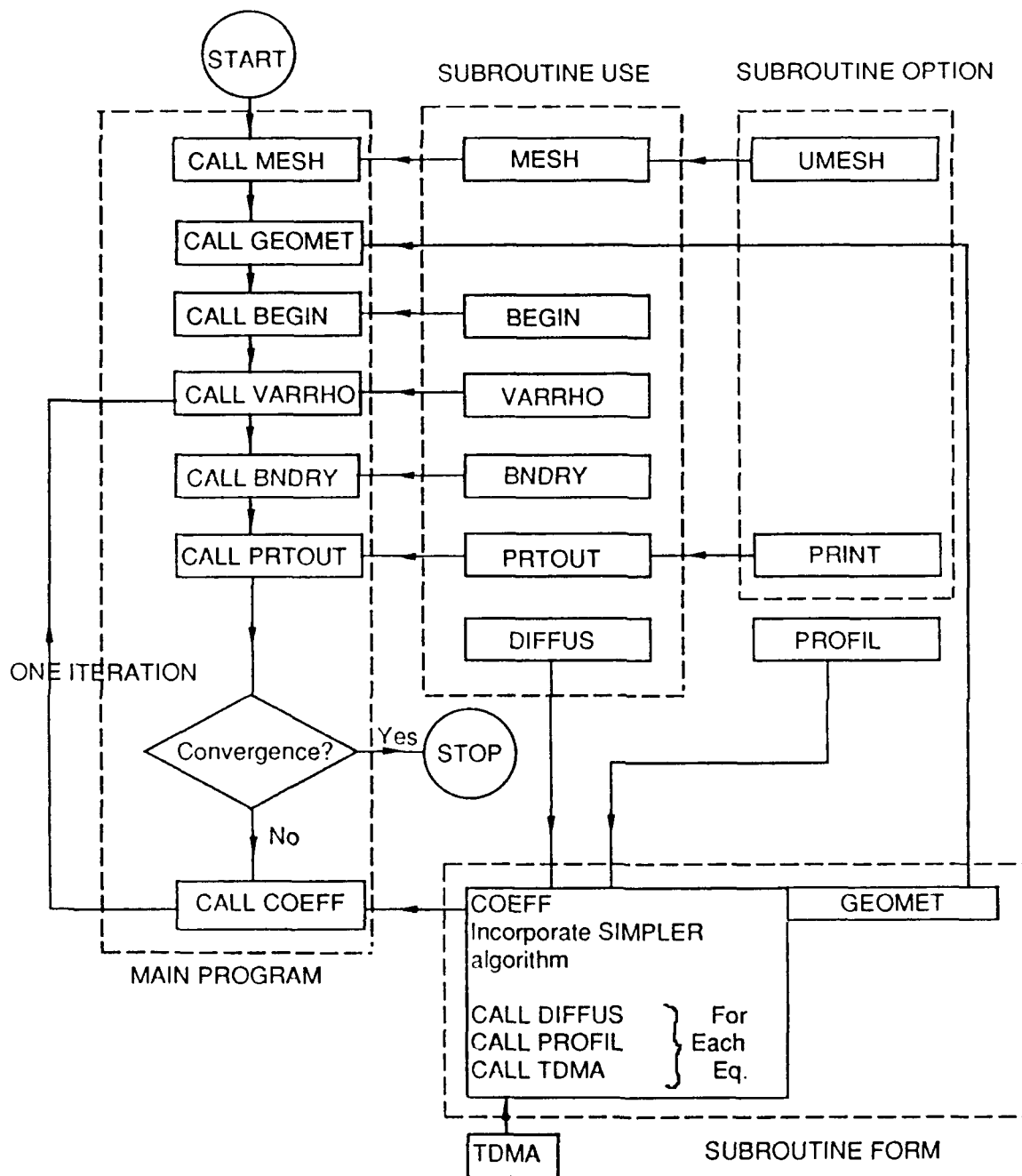


Fig. 1 Program Flow-chart.



## **APPENDIX A - Other FORTRAN Variables in USE**

ITER	Counter for iterations. Program stops when ITER=LAST.
L1,M1,N1	Number of grid points in the X, Y and Z directions, respectively. L1-2, M1-2 and N1-2 are the total number of control volumes in the X, Y and Z directions.
XL,YL,ZL	Domain lengths in the X,Y, and Z directions, respectively.
X(I),Y(J),Z(K)	Locations of the node points where temperatures and pressures are stored. The temperature $T(I,J,K)$ is stored at the location $(X(I),Y(J),Z(K))$ . Note $X(1)=0.,X(L1)=XL$ .
XU(I),YV(J),ZW(K)	Locations of control volume faces. XU(I) is the location of the control volume face perpendicular to the X axis for the control volume around the point $(X(I),Y(J),Z(K))$ such that $XU(I)<X(I)$ . $XU(2)=0.,XU(L1)=XL$ . XU(1) is meaningless. U(I,J,K) is evaluated at the point $(XU(I),Y(J),Z(K))$ . Similar description is applicable for YV(J) and ZW(K).
FX(I), FXM(I)	Interpolation factors in the x direction that enable calculation of a quantity a control volume interface perpendicular to the x direction. EXAMPLE: Temperature at the interface of two control volumes, say at XU(I), can be calculated as $FX(I)*T(I,J,K)+FXM(I)*T(I-1,J,K)$

Similar interpretation for  $FY(J)$ ,  $FYM(J)$ ;  $FZ(K)$ ,  
 $FZM(K)$ .

$XDIF(I), YDIF(J),$

$ZDIF(K)$

$XDIF(I)=X(I)-X(I-1)$ ,  $YDIF(J)=Y(J)-Y(J-1)$ ,

$ZDIF(K)=Z(K)-Z(K-1)$

$XCV(I), YCV(J),$

$ZCV(K)$

Dimensions in the x, y and z directions respectively  
associated with control volume around the point  
( $X(I), Y(J), Z(K)$ ). eg.  $XCV(I)=XU(I+1)-XU(I)$ . Thus  
 $XCV(1)$  and  $XCV(L1)=0$ .

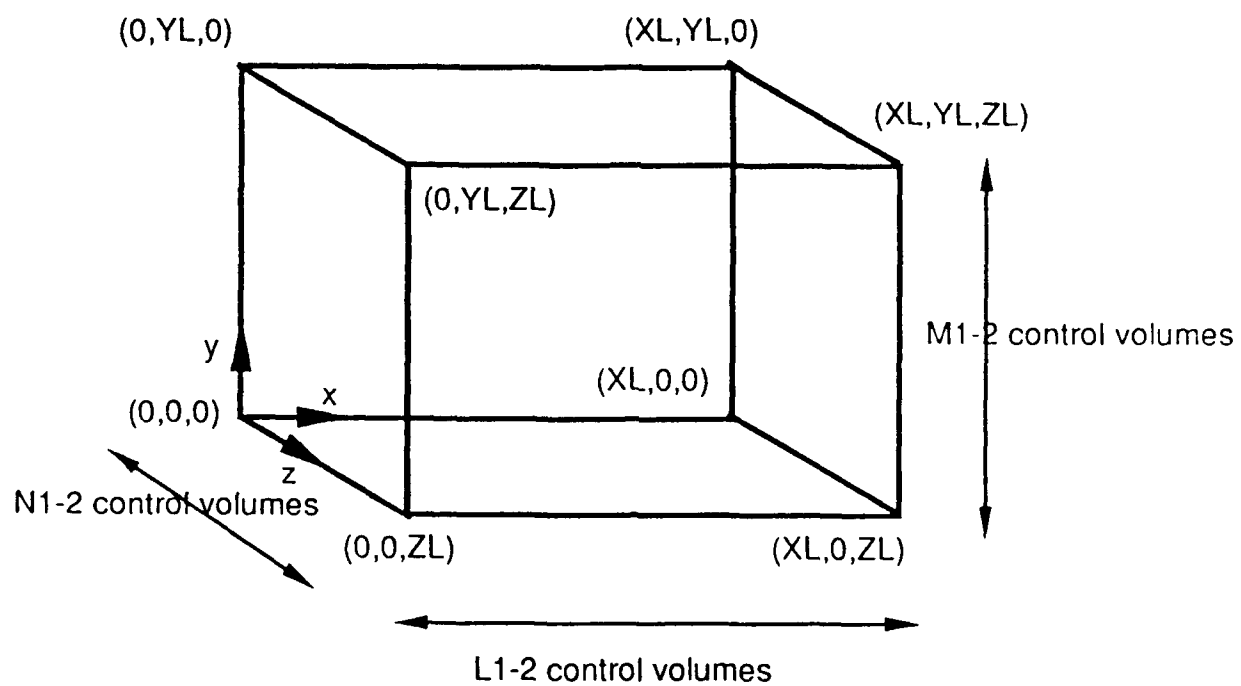


Fig. A-1. The computational domain. The coordinate system is right-handed.

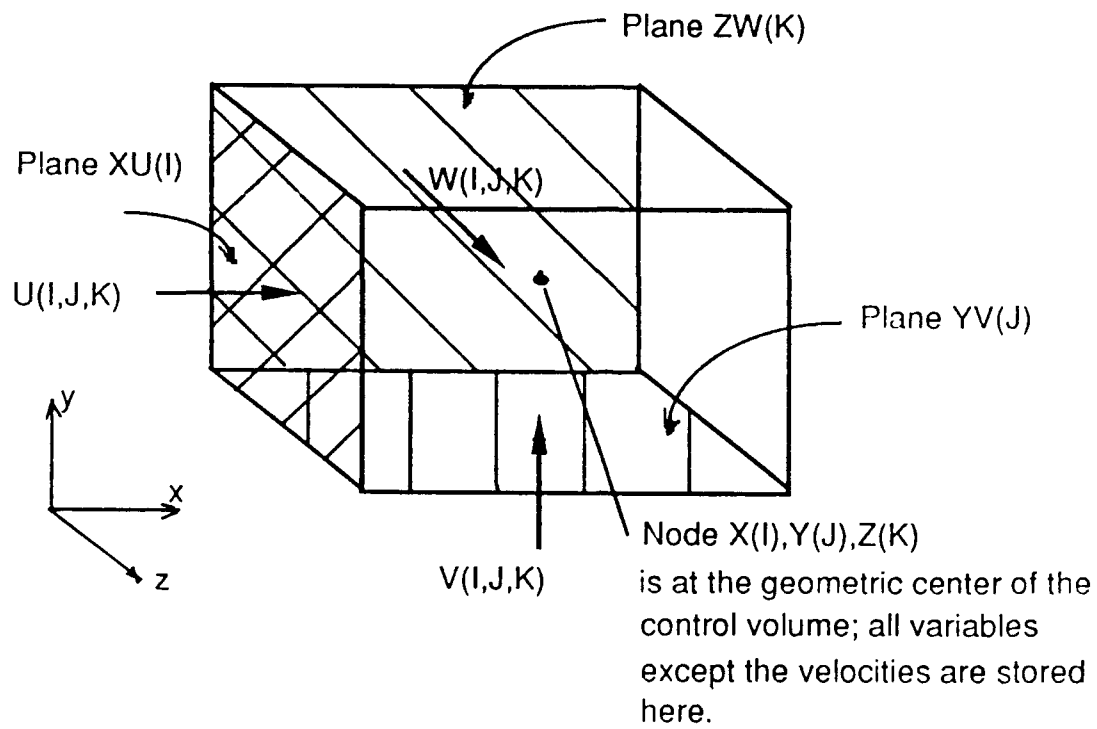


Fig. A-2. The control volume.

**APPENDIX B - Program Source Code. The subroutine USE is for the  
problem described in Appendix C.**



```

COMMON DU(17,17,13),DV(17,17,13),DW(17,17,13),FV(17),FVP(17),
1 FX(17),FXM(17),FY(17),FYM(17),PT(17),QT(17),
2 FZ(17),FZM(17)
COMMON/INDX/RELAX(13),LPRINT(13),LBLK(11),NTIMES(10),
1LSOLVE(10),TIME,DT,XL,YL,ZL,S,RHOCON,ZERO,
2NF,NFMAX,NP,NRHO,NGAM,L1,L2,L3,M1,M2,M3,N1,N2,N3,
3IST,JST,KST,ITER,LAST,
4IPREF,JPREF,KPREF,MODE
DIMENSION D(17),VAR(17),VARM(17),VARP(17),PHIBAR(17)
COMMON/HEADIN/TITLE
CHARACTER*10 TITLE(13)
C*****
ISTF=IST-1
JSTF=JST-1
KSTF=KST-1
IT1=L2+IST
IT2=L3+IST
JT1=M2+JST
JT2=M3+JST
KT1=N2+KST
KT2=N3+KST
C-----
NTSSS=NTIMES(NF)
DO 999 NT=1,NTSSS
DO 391 N=NF,NF
C
IF(.NOT.LBLK(NF))GO TO 60
60 CONTINUE
COMMENCE TDMA LINE BY LINE SWEEPS FOR SOLUTION
DO 90 K=KST,N2
DO 90 J=JST,M2
PT(ISTF)=0.
QT(ISTF)=F(ISTF,J,K,N)
DO 70 I=IST,L2
DENOM=AP(I,J,K)-PT(I-1)*AIM(I,J,K)
PT(I)=AIP(I,J,K)/DENOM
TEMP=CON(I,J,K)+AJP(I,J,K)*F(I,J+1,K,N)+AJM(I,J,K)*F(I,J-1,K,N)
1 +AKP(I,J,K)*F(I,J,K+1,N)+AKM(I,J,K)*F(I,J,K-1,N)
QT(I)=(TEMP+AIM(I,J,K)*QT(I-1))/DENOM
70 CONTINUE
DO 80 II=IST,L2
I=IT1-II
80 F(I,J,K,N)=F(I+1,J,K,N)*PT(I)+QT(I)
90 CONTINUE
C-----
DO 190 KK=KST,N3
K=KT2-KK
DO 190 JJ=JST,M3
J=JT2-JJ
PT(ISTF)=0.
QT(ISTF)=F(ISTF,J,K,N)
DO 170 I=IST,L2
DENOM=AP(I,J,K)-PT(I-1)*AIM(I,J,K)
PT(I)=AIP(I,J,K)/DENOM
TEMP=CON(I,J,K)+AJP(I,J,K)*F(I,J+1,K,N)+AJM(I,J,K)*F(I,J-1,K,N)
1 +AKP(I,J,K)*F(I,J,K+1,N)+AKM(I,J,K)*F(I,J,K-1,N)
QT(I)=(TEMP+AIM(I,J,K)*QT(I-1))/DENOM
170 CONTINUE
DO 180 II=IST,L2
I=IT1-II

```



```

180 F(I,J,K,N)=F(I+1,J,K,N)*PT(I)+QT(I)
190 CONTINUE
C-----
DO 290 I=IST,L2
DO 290 K=KST,N2
PT(JSTF)=0.
QT(JSTF)=F(I,JSTF,K,N)
DO 270 J=JST,M2
DENOM=AP(I,J,K)-PT(J-1)*AJM(I,J,K)
PT(J)=AJP(I,J,K)/DENOM
TEMP=CON(I,J,K)+AKP(I,J,K)*F(I,J,K+1,N)+AKM(I,J,K)*F(I,J,K-1,N)
1 +AIP(I,J,K)*F(I+1,J,K,N)+AIM(I,J,K)*F(I-1,J,K,N)
QT(J)=(TEMP+AJM(I,J,K)*QT(J-1))/DENOM
270 CONTINUE
DO 280 JJ=JST,M2
J=JT1-JJ
280 F(I,J,K,N)=F(I,J+1,K,N)*PT(J)+QT(J)
290 CONTINUE
C-----
DO 390 II=IST,L3
I=IT2-II
DO 390 KK=KST,N3
K=KT2-KK
PT(JSTF)=0.
QT(JSTF)=F(I,JSTF,K,N)
DO 370 J=JST,M2
DENOM=AP(I,J,K)-PT(J-1)*AJM(I,J,K)
PT(J)=AJP(I,J,K)/DENOM
TEMP=CON(I,J,K)+AKP(I,J,K)*F(I,J,K+1,N)+AKM(I,J,K)*F(I,J,K-1,N)
1 +AIP(I,J,K)*F(I+1,J,K,N)+AIM(I,J,K)*F(I-1,J,K,N)
QT(J)=(TEMP+AJM(I,J,K)*QT(J-1))/DENOM
370 CONTINUE
DO 380 JJ=JST,M2
J=JT1-JJ
380 F(I,J,K,N)=F(I,J+1,K,N)*PT(J)+QT(J)
390 CONTINUE
C-----
DO 490 J=JST,M2
DO 490 I=IST,L2
PT(KSTF)=0.
QT(KSTF)=F(I,J,KSTF,N)
DO 470 K=KST,N2
DENOM=AP(I,J,K)-PT(K-1)*AKM(I,J,K)
PT(K)=AKP(I,J,K)/DENOM
TEMP=CON(I,J,K)+AIP(I,J,K)*F(I+1,J,K,N)+AIM(I,J,K)*F(I-1,J,K,N)
1 +AJP(I,J,K)*F(I,J+1,K,N)+AJM(I,J,K)*F(I,J-1,K,N)
QT(K)=(TEMP+AKM(I,J,K)*QT(K-1))/DENOM
470 CONTINUE
DO 480 KK=KST,N2
K=KT1-KK
480 F(I,J,K,N)=F(I,J,K+1,N)*PT(K)+QT(K)
490 CONTINUE
C-----
DO 590 JJ=JST,M3
J=JT2-JJ
DO 590 II=IST,L3
I=IT2-II
PT(KSTF)=0.
QT(KSTF)=F(I,J,KSTF,N)
DO 570 K=KST,N2

```

```

DENOM=AP(I,J,K)-PT(K-1)*AKM(I,J,K)
PT(K)=AKP(I,J,K)/DENOM
TEMP=CON(I,J,K)+AIP(I,J,K)*F(I+1,J,K,N)+AIM(I,J,K)*F(I-1,J,K,N)
1      +AJP(I,J,K)*F(I,J+1,K,N)+AJM(I,J,K)*F(I,J-1,K,N)
QT(K)=(TEMP+AKM(I,J,K)*QT(K-1))/DENOM
570 CONTINUE
DO 580 KK=KST,N2
K=KT1-KK
580 F(I,J,K,N)=F(I,J,K+1,N)*PT(K)+QT(K)
590 CONTINUE
C-----
391 CONTINUE
C*****
999 CONTINUE
C
C*****
ENTRY RESET
C*****
DO 400 K=2,N2
DO 400 J=2,M2
DO 400 I=2,L2
CON(I,J,K)=0
AP(I,J,K)=0
400 CONTINUE
RETURN
END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE FORM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
LOGICAL LSOLVE,LPRINT,LBLK,LSTOP
COMMON F(17,17,13,5),P(17,17,13),RHO(17,17,13),GAM(17,17,13),
1 CON(17,17,13),AKP(17,17,13),AKM(17,17,13),AP(17,17,13),
2 AIP(17,17,13),AIM(17,17,13),AJP(17,17,13),AJM(17,17,13)
COMMON
3 X(17),XU(17),XDIF(17),XCV(17),XCVS(17),
4 Y(17),YV(17),YDIF(17),YCV(17),YCVS(17),
5 Z(17),ZW(17),ZDIF(17),ZCV(17),ZCVS(17),
6 YCVR(17),YCVRS(17),ARX(17),ARXJ(17),ARXJP(17),
7 R(17),RMN(17),SX(17),SXMN(17),XCVI(17),XCVIP(17),
8 YCVJ(17),YCVJP(17),ZCVK(17),ZCVKP(17)
COMMON DU(17,17,13),DV(17,17,13),DW(17,17,13),FV(17),FVP(17),
1 FX(17),FXM(17),FY(17),FYM(17),PT(17),QT(17),
2 FZ(17),FZM(17)
COMMON/INDX/RELAX(13),LPRINT(13),LBLK(11),NTIMES(10),
1LSOLVE(10),TIME,DT,XL,YL,ZL,S,RHOCON,ZERO,
2NF,NFMAX,NP,NRHO,NGAM,L1,L2,L3,M1,M2,M3,N1,N2,N3,
3IST,JST,KST,ITER,LAST,
4IPREF,JPREF,KPREF,MODE
COMMON/HEADIN/TITLE
CHARACTER*10 TITLE(13)
COMMON/CNTL/LSTOP
COMMON/SORC/SMAX,SSUM
COMMON/COEF/FLOW,DIFF,ACOF
DIMENSION U(17,17,13),V(17,17,13),W(17,17,13),PC(17,17,13)
DIMENSION T(17,17,13)
EQUIVALENCE(F(1,1,1,1),U(1,1,1)),(F(1,1,1,2),V(1,1,1)),
1      (F(1,1,1,3),W(1,1,1)),(F(1,1,1,4),PC(1,1,1))
2      ,(F(1,1,1,5),T(1,1,1))
DIMENSION COF1(17,17,13,7),COF2(17,17,13,7),COF3(17,17,13,7)

```

```

        DIMENSION COF4(17,17,13,7),CON1(17,17,13),CON2(17,17,13)
        DIMENSION CON3(17,17,13)
1   FORMAT(15X,'COMPUTATION  IN  CARTESIAN  COORDINATES')
2   FORMAT(15X,'COMPUTATION FOR AXISYMMETRIC SITUATION')
3   FORMAT(15X,'COMPUTATION  IN  POLAR  COORDINATES')
4   FORMAT(14X,40(1H*),//)
C
COME HERE TO CALCULATE GRIDS SPECIFICATION
C*****
ENTRY GEOMET
C*****
C
        L2=L1-1
        L3=L2-1
        M2=M1-1
        M3=M2-1
        N2=N1-1
        N3=N2-1
        X(1)=XU(2)
        DO 5 I=2,L2
5   X(I)=0.5*(XU(I+1)+XU(I))
        X(L1)=XU(L1)
        Y(1)=YV(2)
        DO 10 J=2,M2
10  Y(J)=0.5*(YV(J+1)+YV(J))
        Y(M1)=YV(M1)
        Z(1)=ZW(2)
        DO 7 K=2,N2
7   Z(K)=0.5*(ZW(K+1)+ZW(K))
        Z(N1)=ZW(N1)
C-----
        DO 15 I=2,L1
15  XDIF(I)=X(I)-X(I-1)
        DO 18 I=2,L2
18  XCV(I)=XU(I+1)-XU(I)
        DO 20 I=3,L2
20  XCVS(I)=XDIF(I)
        XCVS(3)=XCVS(3)+XDIF(2)
        XCVS(L2)=XCVS(L2)+XDIF(L1)
        DO 22 I=3,L3
        XCVI(I)=0.5*XCV(I)
22  XCVIP(I)=XCVI(I)
        XCVIP(2)=XCV(2)
        XCVI(L2)=XCV(L2)
        DO 175 K=2,N1
175 ZDIF(K)=Z(K)-Z(K-1)
        DO 178 K=2,N2
178 ZCV(K)=ZW(K+1)-ZW(K)
        DO 270 K=3,N2
270 ZCVS(K)=ZDIF(K)
        ZCVS(3)=ZCVS(3)+ZDIF(2)
        ZCVS(N2)=ZCVS(N2)+ZDIF(N1)
        DO 272 K=3,N3
        ZCVK(K)=0.5*ZCV(K)
272 ZCVKP(K)=ZCVK(K)
        ZCVKP(2)=ZCV(2)
        ZCVK(N2)=ZCV(N2)
        DO 35 J=2,M1
35  YDIF(J)=Y(J)-Y(J-1)
        DO 40 J=2,M2

```

```

40 YCV(J)=YV(J+1)-YV(J)
DO 45 J=3,M2
45 YCVS(J)=YDIF(J)
YCVS(3)=YCVS(3)+YDIF(2)
YCVS(M2)=YCVS(M2)+YDIF(M1)
DO 277 J=3,M3
YCVJ(J)=0.5*YCV(J)
277 YCVJP(J)=YCVJ(J)
YCVJP(2)=YCV(2)
YCVJ(M2)=YCV(M2)
IF(MODE.NE.1) GO TO 55
DO 52 J=1,M1
RMN(J)=1.0
52 R(J)=1.0
GO TO 56
55 DO 50 J=2,M1
50 R(J)=R(J-1)+YDIF(J)
RMN(2)=R(1)
DO 60 J=3,M2
60 RMN(J)=RMN(J-1)+YCV(J-1)
RMN(M1)=R(M1)
56 CONTINUE
DO 57 J=1,M1
SX(J)=1.
SXMN(J)=1.
IF(MODE.NE.3) GO TO 57
SX(J)=R(J)
IF(J.NE.1) SXMN(J)=RMN(J)
57 CONTINUE
DO 62 J=2,M2
YCVR(J)=R(J)*YCV(J)
ARX(J)=YCVR(J)
IF(MODE.NE.3) GO TO 62
ARX(J)=YCV(J)
62 CONTINUE
DO 64 J=4,M3
64 YCVRS(J)=0.5*(R(J)+R(J-1))*YDIF(J)
YCVRS(3)=0.5*(R(3)+R(1))*YCVS(3)
YCVRS(M2)=0.5*(R(M1)+R(M3))*YCVS(M2)
IF(MODE.NE.2) GO TO 67
DO 65 J=3,M3
ARXJ(J)=0.25*(1.+RMN(J)/R(J))*ARX(J)
65 ARXJP(J)=ARX(J)-ARXJ(J)
GO TO 68
67 DO 66 J=3,M3
ARXJ(J)=0.5*ARX(J)
66 ARXJP(J)=ARXJ(J)
68 ARXJP(2)=ARX(2)
ARXJ(M2)=ARX(M2)
DO 70 J=3,M3
FV(J)=ARXJP(J)/ARX(J)
70 FVP(J)=1.-FV(J)
DO 85 I=3,L2
FX(I)=0.5*XCV(I-1)/XDIF(I)
85 FXM(I)=1.-FX(I)
FX(2)=0.
FXM(2)=1.
FX(L1)=1.
FXM(L1)=0.
DO 90 J=3,M2

```

```

    FY(J)=0.5*YCV(J-1)/YDIF(J)
90 FYM(J)=1.-FY(J)
    FY(2)=0.
    FYM(2)=1.
    FY(M1)=1.
    FYM(M1)=0.
    DO 87 K=3,N2
    FZ(K)=0.5*ZCV(K-1)/ZDIF(K)
87 FZM(K)=1.-FZ(K)
    FZ(2)=0.
    FZM(2)=1.
    FZ(N1)=1.
    FZM(N1)=0.
CON,AP,U,V,RHO,PC AND P ARRAYS ARE INITIALIZED HERE
    DO 95 K=1,N1
    DO 95 J=1,M1
    DO 95 I=1,L1
    PC(I,J,K)=0.
    U(I,J,K)=0.
    V(I,J,K)=0.
    W(I,J,K)=0.
    CON(I,J,K)=0.
    AP(I,J,K)=0.
    RHO(I,J,K)=RHOCON
    P(I,J,K)=0.
95 CONTINUE
    IF(MODE.EQ.1) PRINT 1
    IF(MODE.EQ.2) PRINT 2
    IF(MODE.EQ.3) PRINT 3
    PRINT 4
    RETURN

C
COME HERE TO CALCUALTE COEFFICIENTS FOR FINITE DIFF. EQNS.
C*****
    ENTRY COEFF
C*****
C
COEFFICIENTS FOR THE U EQUATION
C
    CALL RESET
    NF=1
    IF(.NOT.LSOLVE(NF)) GO TO 100
    IST=3
    JST=2
    KST=2
    CALL DIFFUS
    REL=1.-RELAX(NF)
    DO 103 I=3,L2
    DO 103 J=2,M2
    DO 103 K=2,N2
C
COEFFICIENTS AEAST AND AWEST
    FL=U(I,J,K)*(FX(I)*RHO(I,J,K)+FXM(I)*RHO(I-1,J,K))
    FLP=U(I+1,J,K)*(FX(I+1)*RHO(I+1,J,K)+FXM(I+1)*RHO(I,J,K))
    FLOW=YCV(J)*ZCV(K)*0.5*(FL+FLP)
    DIFF=YCV(J)*ZCV(K)*GAM(I,J,K)/XCV(I)
    CALL PROFIL
    AIM(I+1,J,K)=ACOF+AMAX1(ZERO, FLOW)
    AIP(I,J,K)=AIM(I+1,J,K)-FLOW
COEFFICIENTS ANORTH AND ASOUTH
    FL=XCVI(I)*V(I,J+1,K)*(FY(J+1)*RHO(I,J+1,K)+FYM(J+1)*RHO(I,J,K))

```

```

    FLM=XCVIP(I-1)*V(I-1,J+1,K)*(FY(J+1)*RHO(I-1,J+1,K)+FYM(J+1)*
1  RHO(I-1,J,K))
    GM=GAM(I,J,K)*GAM(I,J+1,K)
1  /(YCV(J)*GAM(I,J+1,K)+YCV(J+1)*GAM(I,J,K)+
2  1.0E-30)*XCVI(I)
    GMM=GAM(I-1,J,K)*GAM(I-1,J+1,K)
1  /(YCV(J)*GAM(I-1,J+1,K)+YCV(J+1)*
2  GAM(I-1,J,K)+1.E-30)*XCVIP(I-1)
    DIFF=ZCV(K)*2.*(GM+GMM)
    FLOW=ZCV(K)*(FL+FLM)
    CALL PROFIL
    AJM(I,J+1,K)=ACOF+AMAX1(ZERO, FLOW)
    AJP(I,J,K)=AJM(I,J+1,K)-FLOW
COEFFICIENTS AIN AND AOUT
    FL=XCVI(I)*W(I,J,K+1)*(FZ(K+1)*RHO(I,J,K+1)+FZM(K+1)*RHO(I,J,K))
    FLM=XCVIP(I-1)*W(I-1,J,K+1)*(FZ(K+1)*RHO(I-1,J,K+1)+FZM(K+1)*
1  RHO(I-1,J,K))
    GM=GAM(I,J,K)*GAM(I,J,K+1)
1  /(ZCV(K)*GAM(I,J,K+1)+ZCV(K+1)*GAM(I,J,K)+
2  1.0E-30)*XCVI(I)
    GMM=GAM(I-1,J,K)*GAM(I-1,J,K+1)
1  /(ZCV(K)*GAM(I-1,J,K+1)+ZCV(K+1)*
2  GAM(I-1,J,K)+1.E-30)*XCVIP(I-1)
    DIFF=YCV(J)*2.*(GM+GMM)
    FLOW=YCV(J)*(FL+FLM)
    CALL PROFIL
    AKM(I,J,K+1)=ACOF+AMAX1(ZERO, FLOW)
    AKP(I,J,K)=AKM(I,J,K+1)-FLOW
103  CONTINUE
    DO 104 J=2,M2
    DO 104 K=2,N2
COEFFICIENTS AEAST AND AWEST
    AREA=YCV(J)*ZCV(K)
    FLOW=AREA*RHO(1,J,K)*U(2,J,K)
    DIFF=AREA*GAM(1,J,K)/XCV(2)
    CALL PROFIL
    AIM(3,J,K)=ACOF+AMAX1(ZERO, FLOW)
    FLOW=AREA*RHO(L1,J,K)*U(L1,J,K)
    DIFF=AREA*GAM(L1,J,K)/XCV(L2)
    CALL PROFIL
    AIP(L2,J,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
104  CONTINUE
    DO 105 I=3,L2
    DO 105 K=2,N2
COEFFICIENTS ANORTH AND ASOUTH
    FL=XCVI(I)*V(I,2,K)*RHO(I,1,K)
    FLM=XCVIP(I-1)*V(I-1,2,K)*RHO(I-1,1,K)
    FLOW=ZCV(K)*(FL+FLM)
    GM=XCVI(I)*GAM(I,1,K)+XCVIP(I-1)*GAM(I-1,1,K)
    DIFF=ZCV(K)*GM/YDIF(2)
    CALL PROFIL
    AJM(I,2,K)=ACOF+AMAX1(ZERO, FLOW)
    FL=XCVI(I)*V(I,M1,K)*RHO(I,M1,K)
    FLM=XCVIP(I-1)*V(I-1,M1,K)*RHO(I-1,M1,K)
    FLOW=ZCV(K)*(FL+FLM)
    GM=XCVI(I)*GAM(I,M1,K)+XCVIP(I-1)*GAM(I-1,M1,K)
    DIFF=ZCV(K)*GM/YDIF(M1)
    CALL PROFIL
    AJP(I,M2,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
105  CONTINUE

```

```

DO 106 I=3,L2
DO 106 J=2,M2
COEFFICIENTS AIN AND AOUT
FL=XCVI(I)*W(I,J,2)*RHO(I,J,1)
FLM=XCVIP(I-1)*W(I-1,J,2)*RHO(I-1,J,1)
FLOW=YCV(J)*(FL+FLM)
GM=XCVI(I)*GAM(I,J,1)+XCVIP(I-1)*GAM(I-1,J,1)
DIFF=YCV(J)*GM/ZDIF(2)
CALL PROFIL
AKM(I,J,2)=ACOF+AMAX1(ZERO, FLOW)
FL=XCVI(I)*W(I,J,N1)*RHO(I,J,N1)
FLM=XCVIP(I-1)*W(I-1,J,N1)*RHO(I-1,J,N1)
FLOW=YCV(J)*(FL+FLM)
GM=XCVI(I)*GAM(I,J,N1)+XCVIP(I-1)*GAM(I-1,J,N1)
DIFF=YCV(J)*GM/ZDIF(N1)
CALL PROFIL
AKP(I,J,N2)=ACOF+AMAX1(ZERO, FLOW)-FLOW
106 CONTINUE
DO 107 I=3,L2
DO 107 J=2,M2
DO 107 K=2,N2
VOL=YCV(J)*XCVS(I)*ZCV(K)
APT=(RHO(I,J,K)*XCVI(I)+RHO(I-1,J,K)*XCVIP(I-1))
1/(XCVS(I)*DT)
AP(I,J,K)=AP(I,J,K)-APT
CON(I,J,K)=CON(I,J,K)+APT*U(I,J,K)
AP(I,J,K)=
1 (-AP(I,J,K)*VOL+AIP(I,J,K)+AIM(I,J,K)+AJP(I,J,K)+AJM(I,J,K)
2 +AKP(I,J,K)+AKM(I,J,K))
3/RELAX(NF)
CON(I,J,K)=CON(I,J,K)*VOL+REL*AP(I,J,K)*U(I,J,K)
DU(I,J,K)=VOL/XDIF(I)
DU(I,J,K)=DU(I,J,K)/AP(I,J,K)
107 CONTINUE
DO 7099 I=1,L1
DO 7099 J=1,M1
DO 7099 K=1,N1
COF1(I,J,K,1)=AIP(I,J,K)
COF1(I,J,K,2)=AIM(I,J,K)
COF1(I,J,K,3)=AJP(I,J,K)
COF1(I,J,K,4)=AJM(I,J,K)
COF1(I,J,K,5)=AKP(I,J,K)
COF1(I,J,K,6)=AKM(I,J,K)
COF1(I,J,K,7)=AP(I,J,K)
CON1(I,J,K)=CON(I,J,K)
7099 CONTINUE
C TEMPORARY USE OF PC(I,J) TO STORE UHAT
DO 151 K=2,N2
DO 151 J=2,M2
DO 151 I=3,L2
151 PC(I,J,K)=(AIP(I,J,K)*U(I+1,J,K)+AIM(I,J,K)*U(I-1,J,K)
1 +AJP(I,J,K)*U(I,J+1,K)+AJM(I,J,K)*U(I,J-1,K)
2 +AKP(I,J,K)*U(I,J,K+1)+AKM(I,J,K)*U(I,J,K-1)
3 +CON(I,J,K))/AP(I,J,K)
100 CONTINUE
C
COEFFICIENTS FOR THE V EQUATION-----
C
CALL RESET
NF=2

```

```

IF(.NOT.LSOLVE(NF)) GO TO 200
IST=2
JST=3
KST=2
CALL DIFFUS
REL=1.-RELAX(NF)
DO 203 J=3,M2
DO 203 I=2,L2
DO 203 K=2,N2
COEFFICIENTS ANORTH AND ASOUTH
FL=V(I,J,K)*(FY(J)*RHO(I,J,K)+FYM(J)*RHO(I,J-1,K))
FLP=V(I,J+1,K)*(FY(J+1)*RHO(I,J+1,K)+FYM(J+1)*RHO(I,J,K))
FLOW=XCV(I)*ZCV(K)*0.5*(FL+FLP)
DIFF=XCV(I)*ZCV(K)*GAM(I,J,K)/YCV(J)
CALL PROFIL
AJM(I,J+1,K)=ACOF+AMAX1(ZERO, FLOW)
AJP(I,J,K)=AJM(I,J+1,K)-FLOW
COEFFICIENTS AEAST AND AWEST
FL=YCVJ(J)*U(I+1,J,K)*(FX(I+1)*RHO(I+1,J,K)+FXM(I+1)*RHO(I,J,K))
FLM=YCVJP(J-1)*U(I+1,J-1,K)*(FX(I+1)*RHO(I+1,J-1,K)+FXM(I+1)*
1 RHO(I,J-1,K))
GM=GAM(I,J,K)*GAM(I+1,J,K)
1 /(XCV(I)*GAM(I+1,J,K)+XCV(I+1)*GAM(I,J,K)+
2 1.0E-30)*YCVJ(J)
GMM=GAM(I,J-1,K)*GAM(I+1,J-1,K)
1 /(XCV(I)*GAM(I+1,J-1,K)+XCV(I+1)*
2 GAM(I,J-1,K)+1.E-30)*YCVJP(J-1)
DIFF=ZCV(K)*2.*(GM+GMM)
FLOW=ZCV(K)*(FL+FLM)
CALL PROFIL
AIM(I+1,J,K)=ACOF+AMAX1(ZERO, FLOW)
AIP(I,J,K)=AIM(I+1,J,K)-FLOW
COEFFICIENTS AIN AND AOUT
FL=YCVJ(J)*W(I,J,K+1)*(FZ(K+1)*RHO(I,J,K+1)+FZM(K+1)*RHO(I,J,K))
FLM=YCVJP(J-1)*W(I,J-1,K+1)*(FZ(K+1)*RHO(I,J-1,K+1)+FZM(K+1)*
1 RHO(I,J-1,K))
GM=GAM(I,J,K)*GAM(I,J,K+1)
1 /(ZCV(K)*GAM(I,J,K+1)+ZCV(K+1)*GAM(I,J,K)+
2 1.0E-30)*YCVJ(J)
GMM=GAM(I,J-1,K)*GAM(I,J-1,K+1)
1 /(ZCV(K)*GAM(I,J-1,K+1)+ZCV(K+1)*
2 GAM(I,J-1,K)+1.E-30)*YCVJP(J-1)
DIFF=XCV(I)*2.*(GM+GMM)
FLOW=XCV(I)*(FL+FLM)
CALL PROFIL
AKM(I,J,K+1)=ACOF+AMAX1(ZERO, FLOW)
AKP(I,J,K)=AKM(I,J,K+1)-FLOW
203 CONTINUE
DO 204 I=2,L2
DO 204 K=2,N2
COEFFICIENTS ANORTH AND ASOUTH
AREA=XCV(I)*ZCV(K)
FLOW=AREA*RHO(I,1,K)*V(I,2,K)
DIFF=AREA*GAM(I,1,K)/YCV(2)
CALL PROFIL
AJM(I,3,K)=ACOF+AMAX1(ZERO, FLOW)
FLOW=AREA*RHO(I,M1,K)*V(I,M1,K)
DIFF=AREA*GAM(I,M1,K)/YCV(M2)
CALL PROFIL
AJP(I,M2,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW

```



```

204  CONTINUE
      DO 205 J=3,M2
      DO 205 K=2,N2
COEFFICIENTS AEAST AND AWEST
      FL=YCVJ(J)*U(2,J,K)*RHO(1,J,K)
      FLM=YCVJP(J-1)*U(2,J-1,K)*RHO(1,J-1,K)
      FLOW=ZCV(K)*(FL+FLM)
      GM=YCVJ(J)*GAM(1,J,K)+YCVJP(J-1)*GAM(1,J-1,K)
      DIFF=ZCV(K)*GM/XDIF(2)
      CALL PROFIL
      AIM(2,J,K)=ACOF+AMAX1(ZERO, FLOW)
      FL=YCVJ(J)*U(L1,J,K)*RHO(L1,J,K)
      FLM=YCVJP(J-1)*U(L1,J-1,K)*RHO(L1,J-1,K)
      FLOW=ZCV(K)*(FL+FLM)
      GM=YCVJ(J)*GAM(L1,J,K)+YCVJP(J-1)*GAM(L1,J-1,K)
      DIFF=ZCV(K)*GM/XDIF(L1)
      CALL PROFIL
      AIP(L2,J,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
205  CONTINUE
      DO 206 I=2,L2
      DO 206 J=3,M2
COEFFICIENTS AIN AND AOUT
      FL=YCVJ(J)*W(I,J,2)*RHO(I,J,1)
      FLM=YCVJP(J-1)*W(I,J-1,2)*RHO(I,J-1,1)
      FLOW=XCV(I)*(FL+FLM)
      GM=YCVJ(J)*GAM(I,J,1)+YCVJP(J-1)*GAM(I,J-1,1)
      DIFF=XCV(I)*GM/ZDIF(2)
      CALL PROFIL
      AKM(I,J,2)=ACOF+AMAX1(ZERO, FLOW)
      FL=YCVJ(J)*W(I,J,N1)*RHO(I,J,N1)
      FLM=YCVJP(J-1)*W(I,J-1,N1)*RHO(I,J-1,N1)
      FLOW=XCV(I)*(FL+FLM)
      GM=YCVJ(J)*GAM(I,J,N1)+YCVJP(J-1)*GAM(I,J-1,N1)
      DIFF=XCV(I)*GM/ZDIF(N1)
      CALL PROFIL
      AKP(I,J,N2)=ACOF+AMAX1(ZERO, FLOW)-FLOW
206  CONTINUE
      DO 207 I=2,L2
      DO 207 J=3,M2
      DO 207 K=2,N2
      VOL=XCV(I)*YCVS(J)*ZCV(K)
      APT=(RHO(I,J,K)*YCVJ(J)+RHO(I,J-1,K)*YCVJP(J-1))
1 / (YCVS(J)*DT)
      AP(I,J,K)=AP(I,J,K)-APT
      CON(I,J,K)=CON(I,J,K)+APT*V(I,J,K)
      AP(I,J,K)=
1      (-AP(I,J,K)*VOL+AIP(I,J,K)+AIM(I,J,K)+AJP(I,J,K)+AJM(I,J,K)
2      +AKP(I,J,K)+AKM(I,J,K))
3 /RELAX(NF)
      CON(I,J,K)=CON(I,J,K)*VOL+REL*AP(I,J,K)*V(I,J,K)
      DV(I,J,K)=VOL/YDIF(J)
      DV(I,J,K)=DV(I,J,K)/AP(I,J,K)
207  CONTINUE
      DO 8099 I=1,L1
      DO 8099 J=1,M1
      DO 8099 K=1,N1
      COF2(I,J,K,1)=AIP(I,J,K)
      COF2(I,J,K,2)=AIM(I,J,K)
      COF2(I,J,K,3)=AJP(I,J,K)
      COF2(I,J,K,4)=AJM(I,J,K)

```

```
COF2(I,J,K,5)=AKP(I,J,K)
COF2(I,J,K,6)=AKM(I,J,K)
COF2(I,J,K,7)=AP(I,J,K)
CON2(I,J,K)=CON(I,J,K)
```

```
8099 CONTINUE
```

```
200 CONTINUE
```

```
C
COEFFICIENTS FOR THE W EQUATION-----
C
```

```
CALL RESET
NF=3
IF(.NOT.LSOLVE(NF)) GO TO 300
IST=2
JST=2
KST=3
CALL DIFFUS
REL=1.-RELAX(NF)
DO 303 K=3,N2
DO 303 J=2,M2
DO 303 I=2,L2
```

```
COEFFICIENTS AIN AND AOUT
```

```
FL=W(I,J,K)*(FZ(K)*RHO(I,J,K)+FZM(K)*RHO(I,J,K-1))
FLP=W(I,J,K+1)*(FZ(K+1)*RHO(I,J,K+1)+FZM(K+1)*RHO(I,J,K))
FLOW=YCV(J)*XCV(I)*0.5*(FL+FLP)
DIFF=YCV(J)*XCV(I)*GAM(I,J,K)/ZCV(K)
CALL PROFIL
```

```
AKM(I,J,K+1)=ACOF+AMAX1(ZERO, FLOW)
AKP(I,J,K)=AKM(I,J,K+1)-FLOW
```

```
COEFFICIENTS ANORTH AND ASOUTH
```

```
FL=ZCVK(K)*V(I,J+1,K)*(FY(J+1)*RHO(I,J+1,K)+FYM(J+1)*RHO(I,J,K))
FLM=ZCVKP(K-1)*V(I,J+1,K-1)*(FY(J+1)*RHO(I,J+1,K-1)+FYM(J+1)*
1 RHO(I,J,K-1))
GM=GAM(I,J,K)*GAM(I,J+1,K)
1 /(YCV(J)*GAM(I,J+1,K)+YCV(J+1)*GAM(I,J,K)+
2 1.0E-30)*ZCVK(K)
GMM=GAM(I,J,K-1)*GAM(I,J+1,K-1)
1 /(YCV(J)*GAM(I,J+1,K-1)+YCV(J+1)*
2 GAM(I,J,K-1)+1.E-30)*ZCVKP(K-1)
DIFF=XCV(I)*2.*(GM+GMM)
FLOW=XCV(I)*(FL+FLM)
```

```
CALL PROFIL
```

```
AJM(I,J+1,K)=ACOF+AMAX1(ZERO, FLOW)
AJP(I,J,K)=AJM(I,J+1,K)-FLOW
```

```
COEFFICIENTS AEAST AND AWEST
```

```
FL=ZCVK(K)*U(I+1,J,K)*(FX(I+1)*RHO(I+1,J,K)+FXM(I+1)*RHO(I,J,K))
FLM=ZCVKP(K-1)*U(I+1,J,K-1)*(FX(I+1)*RHO(I+1,J,K-1)+FXM(I+1)*
1 RHO(I,J,K-1))
GM=GAM(I,J,K)*GAM(I+1,J,K)
1 /(XCV(I)*GAM(I+1,J,K)+XCV(I+1)*GAM(I,J,K)+
2 1.0E-30)*ZCVK(K)
GMM=GAM(I,J,K-1)*GAM(I+1,J,K-1)
1 /(XCV(I)*GAM(I+1,J,K-1)+XCV(I+1)*
2 GAM(I,J,K-1)+1.E-30)*ZCVKP(K-1)
DIFF=YCV(J)*2.*(GM+GMM)
FLOW=YCV(J)*(FL+FLM)
```

```
CALL PROFIL
```

```
AIM(I+1,J,K)=ACOF+AMAX1(ZERO, FLOW)
AIP(I,J,K)=AIM(I+1,J,K)-FLOW
```

```
303 CONTINUE
```

```
DO 304 J=2,M2
```

```

DO 304 I=2,L2
COEFFICIENTS AIN AND AOUT
AREA=YCV(J)*XCV(I)
FLOW=AREA*RHO(I,J,1)*W(I,J,2)
DIFF=AREA*GAM(I,J,1)/ZCV(2)
CALL PROFIL
AKM(I,J,3)=ACOF+AMAX1(ZERO, FLOW)
FLOW=AREA*RHO(I,J,N1)*W(I,J,N1)
DIFF=AREA*GAM(I,J,N1)/ZCV(N2)
CALL PROFIL
AKP(I,J,N2)=ACOF+AMAX1(ZERO, FLOW)-FLOW
304 CONTINUE
DO 305 I=2,L2
DO 305 K=3,N2
COEFFICIENTS ANORTH AND ASOUTH
FL=ZCVK(K)*V(I,2,K)*RHO(I,1,K)
FLM=ZCVKP(K-1)*V(I,2,K-1)*RHO(I,1,K-1)
FLOW=XCV(I)*(FL+FLM)
GM=ZCVK(K)*GAM(I,1,K)+ZCVKP(K-1)*GAM(I,1,K-1)
DIFF=XCV(I)*GM/YDIF(2)
CALL PROFIL
AJM(I,2,K)=ACOF+AMAX1(ZERO, FLOW)
FL=ZCVK(K)*V(I,M1,K)*RHO(I,M1,K)
FLM=ZCVKP(K-1)*V(I,M1,K-1)*RHO(I,M1,K-1)
FLOW=XCV(I)*(FL+FLM)
GM=ZCVK(K)*GAM(I,M1,K)+ZCVKP(K-1)*GAM(I,M1,K-1)
DIFF=XCV(I)*GM/YDIF(M1)
CALL PROFIL
AJP(I,M2,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
305 CONTINUE
DO 306 K=3,N2
DO 306 J=2,M2
COEFFICIENTS AEAST AND AWEST
FL=ZCVK(K)*U(2,J,K)*RHO(1,J,K)
FLM=ZCVKP(K-1)*U(2,J,K-1)*RHO(1,J,K-1)
FLOW=YCV(J)*(FL+FLM)
GM=ZCVK(K)*GAM(1,J,K)+ZCVKP(K-1)*GAM(1,J,K-1)
DIFF=YCV(J)*GM/XDIF(2)
CALL PROFIL
AIM(2,J,K)=ACOF+AMAX1(ZERO, FLOW)
FL=ZCVK(K)*U(L1,J,K)*RHO(L1,J,K)
FLM=ZCVKP(K-1)*U(L1,J,K-1)*RHO(L1,J,K-1)
FLOW=YCV(J)*(FL+FLM)
GM=ZCVK(K)*GAM(L1,J,K)+ZCVKP(K-1)*GAM(L1,J,K-1)
DIFF=YCV(J)*GM/XDIF(L1)
CALL PROFIL
AIP(L2,J,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
306 CONTINUE
DO 307 I=2,L2
DO 307 J=2,M2
DO 307 K=3,N2
VOL=YCV(J)*ZCVS(K)*XCV(I)
APT=(RHO(I,J,K)*ZCVK(K)+RHO(I,J,K-1)*ZCVKP(K-1))
1/(ZCVS(K)*DT)
AP(I,J,K)=AP(I,J,K)-APT
CON(I,J,K)=CON(I,J,K)+APT*W(I,J,K)
AP(I,J,K)=
1 (-AP(I,J,K)*VOL+AIP(I,J,K)+AIM(I,J,K)+AJP(I,J,K)+AJM(I,J,K)
2 +AKP(I,J,K)+AKM(I,J,K))
3/RELAX(NF)

```

```

CON(I,J,K)=CON(I,J,K)*VOL+REL*AP(I,J,K)*W(I,J,K)
DW(I,J,K)=VOL/ZDIF(K)
DW(I,J,K)=DW(I,J,K)/AP(I,J,K)
307 CONTINUE
DO 9099 I=1,L1
DO 9099 J=1,M1
DO 9099 K=1,N1
COF3(I,J,K,1)=AIP(I,J,K)
COF3(I,J,K,2)=AIM(I,J,K)
COF3(I,J,K,3)=AJP(I,J,K)
COF3(I,J,K,4)=AJM(I,J,K)
COF3(I,J,K,5)=AKP(I,J,K)
COF3(I,J,K,6)=AKM(I,J,K)
COF3(I,J,K,7)=AP(I,J,K)
CON3(I,J,K)=CON(I,J,K)
9099 CONTINUE
300 CONTINUE
C
COEFFICIENTS FOR THE PRESSURE EQUATION
C
NF=NP
IF(.NOT.LSOLVE(NF)) GO TO 500
IST=2
JST=2
KST=2
DO 501 J=2,M2
DO 501 K=2,N2
AIM(2,J,K)=0.0
AIP(L2,J,K)=0.0
CON(2,J,K)=RHO(1,J,K)*U(2,J,K)*YCV(J)*ZCV(K)
501 CONTINUE
DO 502 I=2,L2
DO 502 K=2,N2
AJM(I,2,K)=0.0
AJP(I,M2,K)=0.0
CON(I,2,K)=RHO(I,1,K)*V(I,2,K)*XCV(I)*ZCV(K)
502 CONTINUE
DO 503 I=2,L2
DO 503 J=2,M2
AKM(I,J,2)=0.0
AKP(I,J,N2)=0.0
CON(I,J,2)=RHO(I,J,1)*W(I,J,2)*XCV(I)*YCV(J)
503 CONTINUE
C
DO 504 K=2,N2
DO 504 J=2,M2
DO 504 I=2,L2
AREA=YCV(J)*ZCV(K)
ARHO=AREA*(FX(I+1)*RHO(I+1,J,K)+FXM(I+1)*RHO(I,J,K))
FLOW=ARHO*PC(I+1,J,K)
IF(I.EQ.L2)FLOW=ARHO*U(L1,J,K)
AIP(I,J,K)=ARHO*DU(I+1,J,K)
AIM(I+1,J,K)=AIP(I,J,K)
CON(I,J,K)=CON(I,J,K)-FLOW
CON(I+1,J,K)=CON(I+1,J,K)+FLOW
C
AREA=XCV(I)*ZCV(K)
ARHO=AREA*(FY(J+1)*RHO(I,J+1,K)+FYM(J+1)*RHO(I,J,K))
VHAT = (COF2(I,J+1,K,1)*V(I+1,J+1,K)
1 +COF2(I,J+1,K,2)*V(I-1,J+1,K)

```

```

2      +COF2(I,J+1,K,3)*V(I,J+2,K)
3      +COF2(I,J+1,K,4)*V(I,J,K)
4      +COF2(I,J+1,K,5)*V(I,J+1,K+1)
5      +COF2(I,J+1,K,6)*V(I,J+1,K-1)
6      +CON2(I,J+1,K))/(COF2(I,J+1,K,7)+1.0E-30)

```

```

FLOW=ARHO*VHAT
IF(J.EQ.M2)FLOW=ARHO*V(I,M1,K)
AJP(I,J,K)=ARHO*DV(I,J+1,K)
AJM(I,J+1,K)=AJP(I,J,K)
CON(I,J,K)=CON(I,J,K)-FLOW
CON(I,J+1,K)=CON(I,J+1,K)+FLOW

```

C

```

AREA=XCV(I)*YCV(J)
ARHO=AREA*(FZ(K+1)*RHO(I,J,K+1)+FZM(K+1)*RHO(I,J,K))
WHAT = (COF3(I,J,K+1,1)*W(I+1,J,K+1)
1      +COF3(I,J,K+1,2)*W(I-1,J,K+1)
2      +COF3(I,J,K+1,3)*W(I,J+1,K+1)
3      +COF3(I,J,K+1,4)*W(I,J-1,K+1)
4      +COF3(I,J,K+1,5)*W(I,J,K+2)
5      +COF3(I,J,K+1,6)*W(I,J,K)
6      +CON3(I,J,K+1))/(COF3(I,J,K+1,7)+1.E-30)
FLOW=ARHO*WHAT
IF(K.EQ.N2)FLOW=ARHO*W(I,J,N1)
AKP(I,J,K)=ARHO*DW(I,J,K+1)
AKM(I,J,K+1)=AKP(I,J,K)
CON(I,J,K)=CON(I,J,K)-FLOW
CON(I,J,K+1)=FLOW
AP(I,J,K)=AIP(I,J,K)+AIM(I,J,K)+AJP(I,J,K)+AJM(I,J,K)
1      +AKP(I,J,K)+AKM(I,J,K)

```

504 CONTINUE

```

DO 703 I=1,L1
DO 703 J=1,M1
DO 703 K=1,N1
COF4(I,J,K,1)=AIP(I,J,K)
COF4(I,J,K,2)=AIM(I,J,K)
COF4(I,J,K,3)=AJP(I,J,K)
COF4(I,J,K,4)=AJM(I,J,K)
COF4(I,J,K,5)=AKP(I,J,K)
COF4(I,J,K,6)=AKM(I,J,K)
COF4(I,J,K,7)=AP(I,J,K)

```

703 CONTINUE

```

IF(ITER.LE.1) GO TO 409
DO 408 K=2,N2
DO 408 J=2,M2
DO 408 I=2,L2
AP(I,J,K)=AP(I,J,K)/RELAX(NP)
CON(I,J,K)=CON(I,J,K)+(1.-RELAX(NP))*AP(I,J,K)*P(I,J,K)

```

408 CONTINUE

409 CONTINUE

```

CALL TDMA
NF=1
IST=3
JST=2
KST=2
DO 704 I=1,L1
DO 704 J=1,M1
DO 704 K=1,N1
AIP(I,J,K)=COF1(I,J,K,1)
AIM(I,J,K)=COF1(I,J,K,2)
AJP(I,J,K)=COF1(I,J,K,3)

```

```

    AJM(I,J,K)=COF1(I,J,K,4)
    AKP(I,J,K)=COF1(I,J,K,5)
    AKM(I,J,K)=COF1(I,J,K,6)
    AP(I,J,K)=COF1(I,J,K,7)
    CON(I,J,K)=CON1(I,J,K)
704  CONTINUE
    DO 413 K=2,N2
    DO 413 J=2,M2
    DO 413 I=3,L2
413  CON(I,J,K)=CON(I,J,K)+DU(I,J,K)*AP(I,J,K)*(P(I-1,J,K)-P(I,J,K))
    CALL TDMA
    NF=2
    IST=2
    JST=3
    KST=2
    DO 714 I=1,L1
    DO 714 J=1,M1
    DO 714 K=1,N1
    AIP(I,J,K)=COF2(I,J,K,1)
    AIM(I,J,K)=COF2(I,J,K,2)
    AJP(I,J,K)=COF2(I,J,K,3)
    AJM(I,J,K)=COF2(I,J,K,4)
    AKP(I,J,K)=COF2(I,J,K,5)
    AKM(I,J,K)=COF2(I,J,K,6)
    AP(I,J,K)=COF2(I,J,K,7)
    CON(I,J,K)=CON2(I,J,K)
714  CONTINUE
    DO 513 K=2,N2
    DO 513 I=2,L2
    DO 513 J=3,M2
513  CON(I,J,K)=CON(I,J,K)+DV(I,J,K)*AP(I,J,K)*(P(I,J-1,K)-P(I,J,K))
    CALL TDMA
    NF=3
    IST=2
    JST=2
    KST=3
    DO 724 I=1,L1
    DO 724 J=1,M1
    DO 724 K=1,N1
    AIP(I,J,K)=COF3(I,J,K,1)
    AIM(I,J,K)=COF3(I,J,K,2)
    AJP(I,J,K)=COF3(I,J,K,3)
    AJM(I,J,K)=COF3(I,J,K,4)
    AKP(I,J,K)=COF3(I,J,K,5)
    AKM(I,J,K)=COF3(I,J,K,6)
    AP(I,J,K)=COF3(I,J,K,7)
    CON(I,J,K)=CON3(I,J,K)
724  CONTINUE
    DO 523 J=2,M2
    DO 523 I=2,L2
    DO 523 K=3,N2
523  CON(I,J,K)=CON(I,J,K)+DW(I,J,K)*AP(I,J,K)*(P(I,J,K-1)-P(I,J,K))
    CALL TDMA
C
COEFFICIENTS FOR THE PRESSURE CORRECTION EQUATION-----
C
    DO 706 I=1,L1
    DO 706 J=1,M1
    DO 706 K=1,N1
    AIP(I,J,K)=COF4(I,J,K,1)

```

```

AIM(I,J,K)=COF4(I,J,K,2)
AJP(I,J,K)=COF4(I,J,K,3)
AJM(I,J,K)=COF4(I,J,K,4)
AKP(I,J,K)=COF4(I,J,K,5)
AKM(I,J,K)=COF4(I,J,K,6)
AP(I,J,K)=COF4(I,J,K,7)
706 CONTINUE
NF=4
IF(.NOT.LSOLVE(NF)) GO TO 500
IST=2
JST=2
KST=2
CALL DIFFUS
SMAX=0.
SSUM=0.
DO 410 K=2,N2
DO 410 J=2,M2
DO 410 I=2,L2
VOL=YCV(J)*XCV(I)*ZCV(K)
410 CON(I,J,K)=CON(I,J,K)*VOL
DO 701 K=2,N2
DO 701 J=2,M2
CON(2,J,K)=RHO(1,J,K)*U(2,J,K)*YCV(J)*ZCV(K)
701 CONTINUE
DO 702 I=2,L2
DO 702 K=2,N2
CON(I,2,K)=RHO(I,1,K)*V(I,2,K)*XCV(I)*ZCV(K)
702 CONTINUE
DO 153 I=2,L2
DO 153 J=2,M2
CON(I,J,2)=RHO(I,J,1)*W(I,J,2)*XCV(I)*YCV(J)
153 CONTINUE
C
DO 351 K=2,N2
DO 351 J=2,M2
DO 351 I=2,L2
AREA=YCV(J)*ZCV(K)
ARHO=AREA*(FX(I+1)*RHO(I+1,J,K)+FXM(I+1)*RHO(I,J,K))
FLOW=ARHO*U(I+1,J,K)
CON(I,J,K)=CON(I,J,K)-FLOW
CON(I+1,J,K)=CON(I+1,J,K)+FLOW
C
AREA=XCV(I)*ZCV(K)
ARHO=AREA*(FY(J+1)*RHO(I,J+1,K)+FYM(J+1)*RHO(I,J,K))
FLOW=ARHO*V(I,J+1,K)
CON(I,J,K)=CON(I,J,K)-FLOW
CON(I,J+1,K)=CON(I,J+1,K)+FLOW
C
AREA=XCV(I)*YCV(J)
ARHO=AREA*(FZ(K+1)*RHO(I,J,K+1)+FZM(K+1)*RHO(I,J,K))
FLOW=ARHO*W(I,J,K+1)
CON(I,J,K)=CON(I,J,K)-FLOW
CON(I,J,K+1)=FLOW
PC(I,J,K)=0.
351 CONTINUE
DO 352 I=2,L2
DO 352 J=2,M2
DO 352 K=2,N2
SMAX=AMAX1(SMAX,ABS(CON(I,J,K)))
SSUM=SSUM+CON(I,J,K)

```

```

352  CONTINUE
    CALL TDMA
C
COME HERE TO CORRECT THE VELOCITIES-----
C
    DO 511 K=2,N2
    DO 511 J=2,M2
    DO 511 I=2,L2
    IF(I.NE.2) U(I,J,K)=U(I,J,K)+DU(I,J,K)*(PC(I-1,J,K)-PC(I,J,K))
    IF(J.NE.2) V(I,J,K)=V(I,J,K)+DV(I,J,K)*(PC(I,J-1,K)-PC(I,J,K))
    IF(K.NE.2) W(I,J,K)=W(I,J,K)+DW(I,J,K)*(PC(I,J,K-1)-PC(I,J,K))
511  CONTINUE
500  CONTINUE
C
COEFFICIENTS FOR OTHER EQUATIONS-----
C
    CALL RESET
    IST=2
    JST=2
    KST=2
    DO 600 NF=5,NFMAX
    IF(.NOT.LSOLVE(NF)) GO TO 600
    CALL DIFFUS
    REL=1.-RELAX(NF)
    DO 601 I=2,L2
    DO 601 J=2,M2
    DO 601 K=2,N2
COEFFICIENTS AWEST AND AEAST
    AREA= YCV(J)*ZCV(K)
    FLOW=AREA*U(I+1,J,K)*(FX(I+1)*RHO(I+1,J,K)+FXM(I+1)*RHO(I,J,K))
    DIFF=AREA*2.*GAM(I,J,K)*GAM(I+1,J,K)/(XCV(I)*GAM(I+1,J,K)+
1 XCV(I+1)*GAM(I,J,K)+1.0E-30)
    CALL PROFIL
    AIM(I+1,J,K)=ACOF+AMAX1(ZERO, FLOW)
    AIP(I,J,K)=AIM(I+1,J,K)-FLOW
COEFFICIENTS ANORTH AND ASOUTH
    AREA= XCV(I)*ZCV(K)
    FLOW=AREA*V(I,J+1,K)*(FY(J+1)*RHO(I,J+1,K)+FYM(J+1)*RHO(I,J,K))
    DIFF=AREA*2.*GAM(I,J,K)*GAM(I,J+1,K)/(YCV(J)*GAM(I,J+1,K)+
1 YCV(J+1)*GAM(I,J,K)+1.0E-30)
    CALL PROFIL
    AJM(I,J+1,K)=ACOF+AMAX1(ZERO, FLOW)
    AJP(I,J,K)=AJM(I,J+1,K)-FLOW
COEFFICIENTS AOUT AND AINTO
    AREA= YCV(J)*XCV(I)
    FLOW=AREA*W(I,J,K+1)*(FZ(K+1)*RHO(I,J,K+1)+FZM(K+1)*RHO(I,J,K))
    DIFF=AREA*2.*GAM(I,J,K)*GAM(I,J,K+1)/(ZCV(K)*GAM(I,J,K+1)+
1 ZCV(K+1)*GAM(I,J,K)+1.0E-30)
    CALL PROFIL
    AKM(I,J,K+1)=ACOF+AMAX1(ZERO, FLOW)
    AKP(I,J,K)=AKM(I,J,K+1)-FLOW
601  CONTINUE
    DO 610 J=2,M2
    DO 610 K=2,N2
COEFFICIENTS AWEST AND AEAST
    AREA=YCV(J)*ZCV(K)
    FLOW=AREA*U(2,J,K)*RHO(1,J,K)
    DIFF=AREA*GAM(1,J,K)/XDIF(2)
    CALL PROFIL
    AIM(2,J,K)=ACOF+AMAX1(ZERO, FLOW)

```



```

        FLOW=AREA*U(L1,J,K)*RHO(L1,J,K)
        DIFF=AREA*GAM(L1,J,K)/XDIF(L1)
        CALL PROFIL
        AIP(L2,J,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
610    CONTINUE
        DO 611 I=2,L2
        DO 611 K=2,N2
        COEFFICIENTS ANORTH AND ASOUTH
        AREA= XCV(I)*ZCV(K)
        FLOW=AREA*V(I,2,K)*RHO(I,1,K)
        DIFF=AREA*GAM(I,1,K)/YDIF(2)
        CALL PROFIL
        AJM(I,2,K)=ACOF+AMAX1(ZERO, FLOW)
        FLOW=AREA*V(I,M1,K)*RHO(I,M1,K)
        DIFF=AREA*GAM(I,M1,K)/YDIF(M1)
        CALL PROFIL
        AJP(I,M2,K)=ACOF+AMAX1(ZERO, FLOW)-FLOW
611    CONTINUE
        DO 612 I=2,L2
        DO 612 J=2,M2
        COEFFICIENTS AOUT AND AINTO
        AREA= YCV(J)*XCV(I)
        FLOW=AREA*W(I,J,2)*RHO(I,J,1)
        DIFF=AREA*GAM(I,J,1)/ZDIF(2)
        CALL PROFIL
        AKM(I,J,2)=ACOF+AMAX1(ZERO, FLOW)
        FLOW=AREA*W(I,J,N1)*RHO(I,J,N1)
        DIFF=AREA*GAM(I,J,N1)/ZDIF(N1)
        CALL PROFIL
        AKP(I,J,N2)=ACOF+AMAX1(ZERO, FLOW)-FLOW
612    CONTINUE
        DO 3987 I=1,L1
        DO 3987 J=1,M1
        DO 3987 K=1,N1
        VOL=YCV(J)*XCV(I)*ZCV(K)
        APT=RHO(I,J,K)/DT
        AP(I,J,K)=AP(I,J,K)-APT
        CON(I,J,K)=CON(I,J,K)+APT*F(I,J,K,NF)
        AP(I,J,K)
1      =(-AP(I,J,K)*VOL+AIP(I,J,K)+AIM(I,J,K)+AJP(I,J,K)+AJM(I,J,K)
2      +AKM(I,J,K)+AKP(I,J,K))
3      /RELAX(NF)
        CON(I,J,K)=CON(I,J,K)*VOL+REL*AP(I,J,K)*F(I,J,K,NF)
C      PRINT*,I,J,K
C      PRINT*,AIM(I,J,K),AJM(I,J,K),AKM(I,J,K)
C      PRINT*,AIP(I,J,K),AJP(I,J,K),AKP(I,J,K)
C      PRINT*,AP(I,J,K),CON(I,J,K)
3987    CONTINUE
        CALL TDMA
        600    CONTINUE
        TIME=TIME+DT
        ITER=ITER+1
        IF(ITER.GE.LAST) LSTOP=.TRUE.
        RETURN
        END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        SUBROUTINE OPTION
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        LOGICAL LSOLVE,LPRINT,LBLK,LSTOP
        COMMON F(17,17,13,5),P(17,17,13),RHO(17,17,13),GAM(17,17,13),

```

```

1 CON(17,17,13),AKP(17,17,13),AKM(17,17,13),AP(17,17,13),
2 AIP(17,17,13),AIM(17,17,13),AJP(17,17,13),AJM(17,17,13)
COMMON
3 X(17),XU(17),XDIF(17),XCV(17),XCVS(17),
4 Y(17),YV(17),YDIF(17),YCV(17),YCVS(17),
5 Z(17),ZW(17),ZDIF(17),ZCV(17),ZCVS(17),
6 YCVR(17),YCVRS(17),ARX(17),ARXJ(17),ARXJP(17),
7 R(17),RMN(17),SX(17),SXMN(17),XCVI(17),XCVIP(17),
8 YCVJ(17),YCVJP(17),ZCVK(17),ZCVKP(17)
COMMON DU(17,17,13),DV(17,17,13),DW(17,17,13),FV(17),FVP(17),
1 FX(17),FXM(17),FY(17),FYM(17),PT(17),QT(17),
2 FZ(17),FZM(17)
C COMMON/INDX/NF,NFMAX,NP,NRHO,NGAM,L1,L2,L3,M1,M2,M3,N1,N2,N3,
C 1IST,JST,KST,ITER,LAST,TITLE(13),RELAX(13),TIME,DT,XL,YL,ZL,
C 2IPREF,JPREF,KPREF,LSOLVE(10),LPRINT(13),LBLK(11),MODE,NTIMES(10),
C 3RHOCON,ZERO
COMMON/INDX/RELAX(13),LPRINT(13),LBLK(11),NTIMES(10),
1LSOLVE(10),TIME,DT,XL,YL,ZL,S,RHOCON,ZERO,
2NF,NFMAX,NP,NRHO,NGAM,L1,L2,L3,M1,M2,M3,N1,N2,N3,
3IST,JST,KST,ITER,LAST,
4IPREF,JPREF,KPREF,MODE
COMMON/HEADIN/TITLE
CHARACTER*10 TITLE(13)
COMMON/CNTL/LSTOP
COMMON/SORC/SMAX,SSUM
COMMON/COEF/FLOW,DIFF,ACOF
DIMENSION U(17,17,13),V(17,17,13),W(17,17,13),PC(17,17,13)
DIMENSION T(17,17,13)
EQUIVALENCE(F(1,1,1,1),U(1,1,1)),(F(1,1,1,2),V(1,1,1)),
1 (F(1,1,1,3),W(1,1,1)),(F(1,1,1,4),PC(1,1,1))
2 (F(1,1,1,5),T(1,1,1))
C-----
10 FORMAT(26(1H*),3X,A10,3X,26(1H*))
20 FORMAT(1X,4H I =,I6,6I9)
30 FORMAT(1X,1HJ)
40 FORMAT(1X,I2,3X,1P,7E9.2)
50 FORMAT(1H )
51 FORMAT(1X,'I =',2X,7(I4,5X))
52 FORMAT(1X,'X =',1P,7E9.2)
53 FORMAT('TH =',1P,7E9.2)
54 FORMAT(1X,'J =',2X,7(I4,5X))
55 FORMAT(1X,'Y =',1P,7E9.2)
56 FORMAT(1X,'K =',2X,7(I4,5X))
57 FORMAT(1X,'Z =',1P,7E9.2)
59 FORMAT(1X,'K =',2X,I4)
C*****
ENTRY UMESH
C*****
XU(2)=0.
DX=XL/FLOAT(L1-2)
DO 1 I=3,L1
1 XU(I)=XU(I-1)+DX
YV(2)=0.
DY=YL/FLOAT(M1-2)
DO 2 J=3,M1
2 YV(J)=YV(J-1)+DY
ZW(2)=0.
DZ=ZL/FLOAT(N1-2)
DO 3 K=3,N1
3 ZW(K)=ZW(K-1)+DZ

```

```

      RETURN
C*****
      ENTRY PRINT
C*****
      IF(.NOT.LPRINT(3)) GO TO 80
      80 CONTINUE
C
      IF(.NOT.LPRINT(NP)) GO TO 90
C
CONSTRUCT BOUNDARY PRESSURES BY EXTRAPOLATION
C
      DO 91 K=2,N2
      DO 91 J=2,M2
      P(1,J,K)=(P(2,J,K)*XCVS(3)-P(3,J,K)*XDIF(2))/XDIF(3)
91 P(L1,J,K)=(P(L2,J,K)*XCVS(L2)-P(L3,J,K)*XDIF(L1))/XDIF(L2)
      DO 92 K=2,N2
      DO 92 I=2,L2
      P(I,1,K)=(P(I,2,K)*YCVS(3)-P(I,3,K)*YDIF(2))/YDIF(3)
92 P(I,M1,K)=(P(I,M2,K)*YCVS(M2)-P(I,M3,K)*YDIF(M1))/YDIF(M2)
      DO 93 J=2,M2
      DO 93 I=2,L2
      P(I,J,1)=(P(I,J,2)*ZCVS(3)-P(I,J,3)*ZDIF(2))/ZDIF(3)
93 P(I,J,N1)=(P(I,J,N2)*ZCVS(N2)-P(I,J,N3)*ZDIF(N1))/ZDIF(N2)
      DO 94 K=2,N2
      P(1,1,K)=P(2,1,K)+P(1,2,K)-P(2,2,K)
94 CONTINUE
      DO 95 J=2,M2
      P(1,J,1)=P(2,J,1)+P(1,J,2)-P(2,J,2)
95 CONTINUE
      DO 96 I=2,L2
      P(I,1,1)=P(I,2,1)+P(I,1,2)-P(I,2,2)
96 CONTINUE
      P(1,1,1)=(P(1,1,2)+P(1,2,1)+P(2,1,1))/3.0
      P(L1,1,1)=(P(L2,1,1)+P(L1,2,1)+P(L1,1,2))/3.0
      P(1,1,N1)=(P(1,1,N2)+P(1,2,N1)+P(2,1,N1))/3.0
      P(1,M1,1)=(P(1,M2,1)+P(1,M1,2)+P(2,M1,1))/3.0
      P(L1,M1,1)=(P(L2,M1,1)+P(L1,M2,1)+P(L1,M1,2))/3.0
      P(1,M1,N1)=(P(2,M1,N1)+P(1,M2,N1)+P(1,M1,N2))/3.0
      P(L1,M1,N1)=(P(L2,M1,N1)+P(L1,M2,N1)+P(L1,M1,N2))/3.0
      PREF=P(IPREF,JPREF,KPREF)
      DO 97 K=1,N1
      DO 97 J=1,M1
      DO 97 I=1,L1
97 P(I,J,K)=P(I,J,K)-PREF
90 CONTINUE
C
      PRINT 50
      IEND=0
301 IF(IEND.EQ.L1) GO TO 310
      IBEG=IEND+1
      IEND=IEND+7
      IEND=MIN0(IEND,L1)
      PRINT 50
      PRINT 51,(I,I=IBEG,IEND)
      IF(MODE.EQ.3) GO TO 302
      PRINT 52,(X(I),I=IBEG,IEND)
      GO TO 303
302 PRINT 53,(X(I),I=IBEG,IEND)
303 GO TO 301
310 JEND=0

```



```

2 AIP(17,17,13),AIM(17,17,13),AJP(17,17,13),AJM(17,17,13)
COMMON
3 X(17),XU(17),XDIF(17),XCV(17),XCVS(17),
4 Y(17),YV(17),YDIF(17),YCV(17),YCVS(17),
5 Z(17),ZW(17),ZDIF(17),ZCV(17),ZCVS(17),
6 YCVR(17),YCVRS(17),ARX(17),ARXJ(17),ARXJP(17),
7 R(17),RMN(17),SX(17),SXMN(17),XCVI(17),XCVIP(17),
8 YCVJ(17),YCVJP(17),ZCVK(17),ZCVKP(17)
COMMON DU(17,17,13),DV(17,17,13),DW(17,17,13),FV(17),FVP(17),
1 FX(17),FXM(17),FY(17),FYM(17),PT(17),QT(17),
2 FZ(17),FZM(17)
COMMON/INDX/RELAX(13),LPRINT(13),LBLK(11),NTIMES(10),
1LSOLVE(10),TIME,DT,XL,YL,ZL,S,RHOCON,ZERO,
2NF,NFMAX,NP,NRHO,NGAM,L1,L2,L3,M1,M2,M3,N1,N2,N3,
3IST,JST,KST,ITER,LAST,
4IPREF,JPREF,KPREF,MODE
COMMON/HEADIN/TITLE
CHARACTER*10 TITLE(13)
COMMON/CNTL/LSTOP
COMMON/SORC/SMAX,SSUM
COMMON/COEF/FLOW,DIFF,ACOF
DIMENSION U(17,17,13),V(17,17,13),W(17,17,13),PC(17,17,13)
DIMENSION T(17,17,13)
EQUIVALENCE(F(1,1,1,1),U(1,1,1)),(F(1,1,1,2),V(1,1,1)),
1 (F(1,1,1,3),W(1,1,1)),(F(1,1,1,4),PC(1,1,1))
2 (F(1,1,1,5),T(1,1,1))
DATA NFMAX,NP,NRHO,NGAM/5,6,7,8/
DATA LSTOP,LSOLVE,LPRINT/24*.FALSE./
DATA MODE,TIME,ITER/1,0.,0/
DATA RELAX,NTIMES/13*1.0,10*1/
DATA LBLK/11*.TRUE./
DATA DT,IPREF,JPREF,KPREF,RHOCON/1.E+10,1,1,1,1.0/
C*****
C NF=1,2,3 STAND FOR U,V AND W VELOCITIES.
C NF=6 IS FOR PRESSURE
C NF=5 IS FOR TEMPERATURE
C LSOLVE=TRUE SOLVES THAT PARTICULAR PHI
DATA (LSOLVE(I),I=1,6)/6*.TRUE./
C LPRINT(NF)=TRUE PRINTS VARIABLE ASSOCIATED WITH NF ON CALLING PRINT
DATA LPRINT(5)/1*.TRUE./
C TERMINATE ITERATIONS AT ITER=LAST
DATA LAST/100/
C UNDERRELAXATION FACTORS
DATA RELAX(1),RELAX(2),RELAX(3),RELAX(4)/0.5,0.5,0.5,1.0/
DATA RELAX(5),RELAX(6)/1.0,1.0/
C TITLES FOR THE FIELD PRINTOUTS
DATA TITLE(1),TITLE(2),TITLE(3),TITLE(5),TITLE(6)/
1 'U','V','W','T','P'/
C NUMBER OF SWEEPS IN THE LINE-BY-LINE TDMA ALGORITHM
DATA NTIMES(3),NTIMES(4)/2*5/
DATA ZERO/0./
END
C*****
SUBROUTINE USE
C*****
C IF LARGER NUMBER OF GRID POINTS IS TO BE USED, THE DIMENSION
C STATEMENTS MUST BE CHANGED THROUGH THE PROGRAM TO ACCOMODATE
C VALUES GRAETER THAN (17,17,13) ETC.
C*****
LOGICAL LREAD,LWRITE

```

```

LOGICAL LSOLVE,LPRINT,LBLK,LSTOP
COMMON F(17,17,13,5),P(17,17,13),RHO(17,17,13),GAM(17,17,13),
1 CON(17,17,13),AKP(17,17,13),AKM(17,17,13),AP(17,17,13),
2 AIP(17,17,13),AIM(17,17,13),AJP(17,17,13),AJM(17,17,13)
COMMON
3 X(17),XU(17),XDIF(17),XCV(17),XCVS(17),
4 Y(17),YV(17),YDIF(17),YCV(17),YCVS(17),
5 Z(17),ZW(17),ZDIF(17),ZCV(17),ZCVS(17),
6 YCVR(17),YCVRS(17),ARX(17),ARXJ(17),ARXJP(17),
7 R(17),RMN(17),SX(17),SXMN(17),XCVI(17),XCVIP(17),
8 YCVJ(17),YCVJP(17),ZCVK(17),ZCVKP(17)
COMMON DU(17,17,13),DV(17,17,13),DW(17,17,13),FV(17),FVP(17),
1 FX(17),FXM(17),FY(17),FYM(17),PT(17),QT(17),
2 FZ(17),FZM(17)
COMMON/INDX/RELAX(13),LPRINT(13),LBLK(11),NTIMES(10),
1LSOLVE(10),TIME,DT,XL,YL,ZL,S,RHOCON,ZERO,
2NF,NFMAX,NP,NRHO,NGAM,L1,L2,L3,M1,M2,M3,N1,N2,N3,
3IST,JST,KST,ITER,LAST,
4IPREF,JPREF,KPREF,MODE
COMMON/HEADIN/TITLE
CHARACTER*10 TITLE(13)
COMMON/CNTL/LSTOP
COMMON/SORC/SMAX,SSUM
COMMON/COEF/FLOW,DIFF,ACOF
DIMENSION U(17,17,13),V(17,17,13),W(17,17,13),PC(17,17,13)
DIMENSION T(17,17,13),BUOY(17,17,13)
C SPECIFY EQUIVALENCE FOR QUANTITIES INSIDE AND OUTSIDE SUBROUTINE USE
C T=TEMPERATURE,PC=PRESSURE CORRECTION
EQUIVALENCE(F(1,1,1,1),U(1,1,1)),(F(1,1,1,2),V(1,1,1)),
1 (F(1,1,1,3),W(1,1,1)),(F(1,1,1,4),PC(1,1,1))
2 (F(1,1,1,5),T(1,1,1))
DIMENSION ANUZ(13),ANU3PT(13)
CHARACTER*1 AREAD
C
C*****
ENTRY MESH
C
C DOMAIN LENGHTS IN THE 3 DIRECTIONS
XL=1.0
YL=1.0
ZL=1.0
C NUMBER OF GRID POINTS IN THE 3 DIRECTIONS
L1=17
M1=17
N1=13
C INVOKE THE UNIFORM MESH OPTION
CALL UMESH
RETURN
C
ENTRY BEGIN
C ITERATIONS STOP AT ITER=LAST
WRITE(*,*)'NO. ITERATIONS'
READ(*,*)LAST
C READ RAYLEIGH NUMBER
WRITE(*,*)'RALI='
READ(*,*)RA
C READ PRNDTL NUMBER
WRITE(*,*)'PRANTL='
READ(*,*)PR
C DETERMINE IF WANT TO USE A PREVIOUSLY COMPUTED SOLUTION AS

```

```

C AN INITIAL GUESS
  WRITE(*,*)'READ FROM INPUT FILE (0/1)'
  READ(*,*)IREAD
  IF(IREAD.EQ.1) LREAD=.TRUE.
  IF(IREAD.EQ.0) LREAD=.FALSE.
C PROVIDE INITIAL GUESS. THE PROGRAM SOLVES FOR THE INTERIOR POINTS
C ONLY. HENCE THE BOUNDARY VALUSE FOR THE TEMPERATURES AT THE HOT
C AND COLD BOUNDARIES HAVE BEEN ALREADY SPECIFIED.
  DO 101 I=1,L1
  DO 101 J=1,M1
  DO 101 K=1,N1
  U(I,J,K)=0.
  V(I,J,K)=0.
  W(I,J,K)=0.
  T(I,J,K)= 0.
C HOT WALL AT X=0.
  T(1,J,K)=1.0
101 CONTINUE
  IF(.NOT.LREAD) RETURN
C READ DATA FROM INPUT FILE
  REWIND (4)
  READ(4) X,Y,Z,XU,YV,ZW,U,V,W,P,T
  RETURN

C
C*****
  ENTRY VARRHO
C*****
  RETURN

C
C*****
C INCORPORATE BOUNDARY CONDITIONS
  ENTRY BNDRY
  DO 864 I=1,L1
  DO 864 J=1,M1
C UPDATE VELOCITIES AND TEMPERATURE AT THE SYMMETRY LINE (Z=ZL)
C ACORDING TO THE ZERO GRADIENT BOUNDARY CONDITION FOR U,V AND T
  U(I,J,N1)=U(I,J,N2)
  V(I,J,N1)=V(I,J,N2)
  T(I,J,N1)=T(I,J,N2)
C ADIABATIC SIDE (Z=0)
  T(I,J,1)=T(I,J,2)
864 CONTINUE
  DO 865 I=1,L1
  DO 865 K=1,N1
C ADIABATIC BOTTOM (Y=0)
  T(I,1,K)=T(I,2,K)
C ADIABATIC TOP (Y=YL)
  T(I,M1,K)=T(I,M2,K)
865 CONTINUE
  RETURN

C
  ENTRY PRTOUT
  IF(ITER.NE.0) GO TO 400
  PRINT 401
401 FORMAT (1X,'SIMPLER',//)
400 CONTINUE
C MONITOR SSUM,SMAX AND OTHER QUANTITES AS ITERATIONS PROCEED
C ON CONVERGENCE, SMAX SHOULD BE VERY SMALL (LESS THAN 1.0E-04)
C SSUM SHOLD ACHIEVE A SMALL VALUE WITHIN A FEW ITERATIONS, WELL
C BEFORE CONVERGENCE. SSUM WILL NOT BE SMALL IF THE BOUNDARY

```

```

C   CONDITIONS ARE NOT WRITTEN CORRECTLY.
      PRINT 403,ITER,SSUM,SMAX,U(5,5,9),V(5,5,5),T(8,6,5)
403  FORMAT (I6,5E12.4)
COMPUTE LOCAL AND GLOBAL NUSSELT NUMBERS AS ITERATIONS PROCEED
      IF( (ITER.EQ.250)
        2.OR.(ITER.EQ.500)
        3.OR.(ITER.EQ.750)
        4.OR.(ITER.EQ.1000)
        5.OR.(ITER.EQ.1250)
        6.OR.(ITER.EQ.1500)
        7.OR.(ITER.EQ.1750)
        8.OR.(ITER.EQ.2000)
        1.OR.(ITER.EQ.2250)
        2.OR.(ITER.EQ.2500)
        3.OR.(ITER.EQ.2750)
        4.OR.(ITER.EQ.3000)
        5.OR.(ITER.EQ.LAST)) THEN
      WRITE (9,*)'ITER,SSUM,SMAX,U(5,5,9),V(5,5,5),T(8,6,5)'
      WRITE (9,*)ITER,SSUM,SMAX,U(5,5,9),V(5,5,5),T(8,6,5)
COMPUTE AVERAGE NUSSELT NUMBER
      ANUHOT=0.0
      ANUCLD=0.0
      DO 666 K=1,N1
      ANUZ(K)=0.0
      DO 667 J=1,M1
COMPUTE AVERAGE NUSSELT NUMBER AT THE HOT AND COLD WALL
      ANUHOT=ANUHOT+(T(1,J,K)-T(2,J,K))*YCV(J)*ZCV(K)/(XDIF(2)*YL*ZL)
      ANUCLD=ANUCLD+(T(M2,J,K)-T(M1,J,K))*YCV(J)*ZCV(K)/(XDIF(L1)*YL*ZL)
COMPUTE LOCAL NUSSELT NUMBER AT THE HOT WALL
      ANUZ(K)=ANUZ(K)+(T(1,J,K)-T(2,J,K))*YCV(J)/(XDIF(2)*YL)
      1  *YCV(J)/(XDIF(2)*YL)
667  CONTINUE
C   WRITE NUSSELT NUMBERS TO AN OUTPUT FILE
      WRITE(9,*)'Z(K)=' ,Z(K) , 'ANUZ(K)=' ,ANUZ(K)
666  CONTINUE
      WRITE(9,*)'ANUHOT =' ,ANUHOT
      WRITE(9,*)'ANUCLD =' ,ANUCLD
      IF(ITER.EQ.LAST) WRITE(9,*)'RA=' ,RA , 'PR=' ,PR , 'ZL=' ,ZL
      WRITE(9,*)'*****'
      ENDIF
      IF(ITER.NE.LAST) RETURN
C   GET A FIELD PRINTOUT AFTER ITERATIONS STOP
      CALL PRINT
C   WRITE IN ORDER TO COMMENCE NEW PROBLEM
      REWIND(4)
      WRITE(4) X,Y,Z,XU,YV,ZW,U,V,W,P,T
      RETURN
C*****
      ENTRY DIFFUS
      IF (NF.EQ.4) RETURN
      DO 500 I=1,L1
      DO 500 J=1,M1
      DO 500 K=1,N1
C   DIFFUSIVITY FOR THE U, V OR W EQUATIONS
      GAM(I,J,K)=PR
C   SPECIFY ZERO DIFFUSIVITY FOR THE ZERO GRADIENT CONDITION AT Z=ZL
C   FOR THE U AND V VELOCITIES.
      IF(NF.NE.3)GAM(I,J,N1)=0.0
C   DIFFUSIVITY FOR THE ENERGY EQUATION
      IF (NF.EQ.5) THEN

```



```

      GAM(I,J,K)=1.0
C   SPECIFY ZERO DIFFUSIVITIES FOR THE ADIABATIC BOUNDARIES
      GAM(I,1,K)=0.0
      GAM(I,M1,K)=0.0
      GAM(I,J,N1)=0.0
      GAM(I,J,1)=0.0
      ENDIF
500   CONTINUE
      IF (NF.NE.2) RETURN
C   SOURCE TERMS ARE EVALUATED ONLY FOR INTERIOR POINTS
      DO 501 I=2,L2
      DO 501 J=3,M2
      DO 501 K=2,N2
C   INTERPOLATE TO GET THE VALUE OF TEMPERATURE AT THE CONTROL VOLUME
C   INTERFACE, SINCE THE VELOCITY U IS EVALUATED AT THE INTERFACE.
      TM=FY(J)*T(I,J,K)+FYM(J)*T(I,J-1,K)
      CON1=RA*PR*TM
C   SUPPLY ONLY PART OF THE SOURCE TERM IN THE MOMENTUM EQ. TO AVOID
C   DIVERGENCE BEFORE 200 ITERATIONS
      CON(I,J,K)=FLOAT(ITER**2)*CON1/200.**2
      IF (ITER.GT.200)CON(I,J,K)=CON1
501   CONTINUE
      RETURN
      END

```

?

## **APPENDIX C - Natural convection in a rectangular box.**

The problem being considered is that of natural convection in a fluid-filled rectangular box. Two vertical opposite walls are maintained at a hot and a cold temperature as shown in Fig. C-1. The remaining box walls are perfectly insulated. The objective is to numerically solve for the fluid flow and heat transfer inside the box arising due to buoyancy effects. The governing equations for the steady-state problem assuming laminar flow, constant properties for the fluid, no viscous dissipation and the Boussinesq approximation to be true are:

Continuity

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (C-1)$$

Momentum (x direction)

$$\begin{aligned} \frac{\partial(uu)}{\partial x} + \frac{\partial(vu)}{\partial y} + \frac{\partial(wu)}{\partial z} = \\ - \frac{\partial p}{\partial x} + Pr \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \end{aligned} \quad (C-2)$$

Momentum (y direction)

$$\begin{aligned} \frac{\partial(uv)}{\partial x} + \frac{\partial(vv)}{\partial y} + \frac{\partial(wv)}{\partial z} = \\ - \frac{\partial p}{\partial y} + Pr \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) + RaPr\theta \end{aligned} \quad (C-3)$$

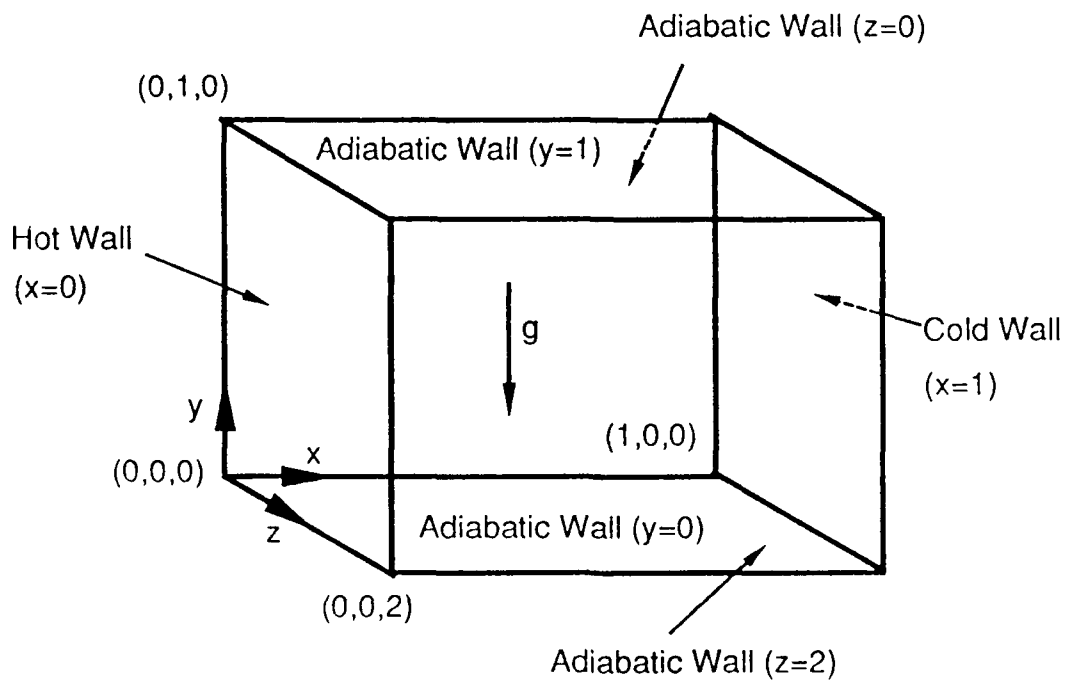


Fig. C-1. Natural convection in a rectangular enclosure.

Momentum (z direction)

$$\frac{\partial(uw)}{\partial x} + \frac{\partial(vw)}{\partial y} + \frac{\partial(w^2)}{\partial z} = -\frac{\partial p}{\partial z} + Pr\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \quad (C-4)$$

Energy

$$\frac{\partial(u\theta)}{\partial x} + \frac{\partial(v\theta)}{\partial y} + \frac{\partial(w\theta)}{\partial z} = \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + \frac{\partial^2 \theta}{\partial z^2} \quad (C-5)$$

The above non-dimensional equations have been derived using the scaling of  $H$  and  $\alpha/H$  for the length and velocity, respectively; where  $H$  is the characteristic dimension of the box in the  $x$  direction and  $\alpha$  is the thermal diffusivity of the fluid.  $Pr = \nu/\alpha$  is the Prandtl number of the fluid and  $Ra = g\beta(T_h - T_c)H^3/\alpha\nu$  is the Rayleigh number;  $\nu$  is the kinematic viscosity,  $g$  is the gravitational acceleration,  $\beta$  is the coefficient of thermal expansion and  $T_h$  and  $T_c$  are the temperatures of the hot and cold walls, respectively. The temperature has been non-dimensionalized as  $\theta = (T - T_c)/(T_h - T_c)$ .

As mentioned earlier, the equations to be solved are cast into the canonical form in Eq. (1). The appropriate definitions for  $\rho$ ,  $\phi$ ,  $\Gamma$ ,  $S_p$  and  $S_c$  are given in Table C-1:

Table C-1 Definition of variables in the canonical equation (Eq. 1)<sup>(1)</sup>

Equation <sup>(2)</sup>	$\phi$	$\Gamma$	$S_p$	$S_c^{(3)}$
x momentum	u	Pr	0	0
y momentum	v	Pr	0	PrRa $\theta$
z momentum	w	Pr	0	0
Energy	$\theta$	1	0	0

(1) Value of  $\rho$  to be used in the canonical form is 1, and is set in variable RHOCON in the **BLOCK DATA**.

(2) It is not required to cast the continuity equation in the canonical form.

(3) The pressure gradient terms are incorporated internally in the program code and need not be specified as source terms in the subroutine USE.

Taking advantage of the symmetry plane  $z=1$ , the computational domain is restricted to  $z \leq 1$ . The appropriate boundary conditions then become:

$$u, v, w, \frac{\partial \theta}{\partial z} = 0 \quad \text{at } z=0$$

$$\frac{\partial u}{\partial z}, \frac{\partial v}{\partial z}, w, \frac{\partial \theta}{\partial z} = 0 \quad \text{at } z=1$$

$$u=v=w=0, \theta=1 \quad \text{at } x=0$$

$$u, v, w, \theta=0 \quad \text{at } x=1$$

$$u, v, w, \frac{\partial \theta}{\partial y} = 0 \quad \text{at } y=0 \text{ and } 1$$

### **Program Implementation:**

As mentioned earlier, the user needs set up only the subroutine USE and BLOCK DATA for a particular problem to be solved. The Subroutine USE and other BLOCK DATA options to be used for this problem are given at the end of the source code in Appendix B. The subroutine is commented appropriately for the user to understand the program usage. Special considerations in the subroutine USE are as follows:

- (1) Whenever a gradient of a dependent variable is specified to be zero at a boundary, the boundary value of  $\Gamma$  associated with that  $\phi$  is set to zero in ENTRY DIFFUS. Thus the dependency of the boundary value of the variable on the solution of the variable at the adjacent node is dropped, leading to much faster convergence. The same solution is obtained if the boundary  $\Gamma$  values were kept as those in Table C-1, however, convergence will be slower.
- (2) For the y momentum equation, the full source term  $RaPr\theta$  is not used right from the start of the iterations, but only a portion of it is used and the source term is gradually increased with iterations to its fullest value. This helped in obtaining convergence, which could not be obtained earlier with the full source term for all iterations. The user is referred to Patankar [1], Ch. 7, for a detailed discussion on convergence and underrelaxation techniques for the source terms.
- (3) Since the u velocity  $U(I,J,K)$  is evaluated on the staggered grid at the grid-point location of the interface of the control volumes around

$(X(I), Y(J), Z(K))$  and  $(X(I), Y(J-1), Z(K))$ , in the source term  $RaPr\theta$ , the temperature  $\theta$  must be evaluated at the interface via interpolation.

**Results:**

Shown in Fig. C-2 is the local Nusselt number at the hot wall compared with the solution from Mallinson and Davis [2]. The local Nusselt number has been defined as:

$$Nu(z) = \int_0^1 - \frac{\partial \theta}{\partial x}(0, y, z) dy \quad (C.6)$$

As can be seen, the agreement is quite good. The small disagreement is attributed to the difference in the numerical procedure, difference in discretization procedure and difference in numerical evaluation of the Nusselt number.



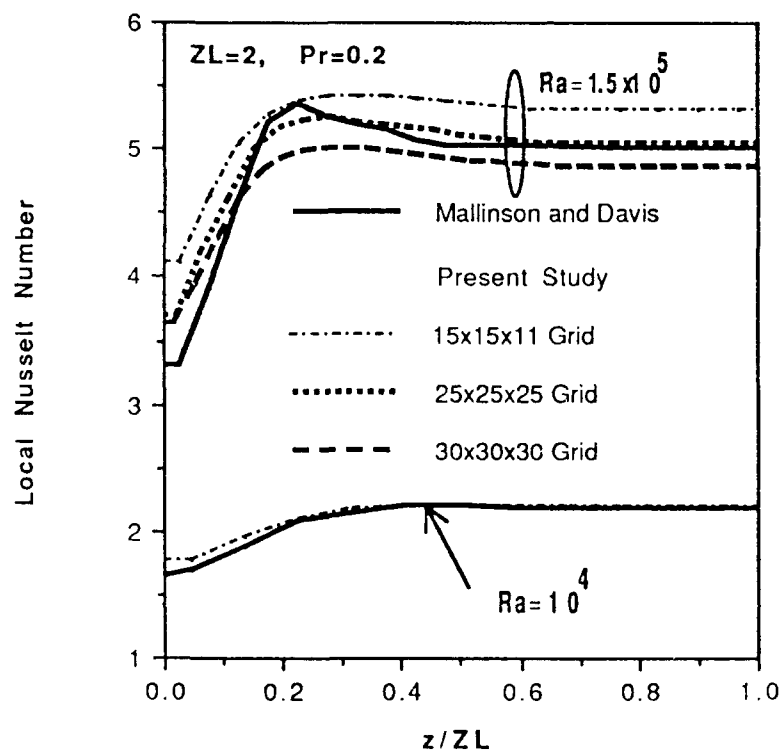


Fig. C-2. Comparison of local Nusselt numbers.

## REFERENCES

1. Patankar, S. V., 1980, *Numerical Heat Transfer and Fluid Flow*, Hemisphere.
2. Mallinson, G. D. and Davis, G. De Vahl, 1977, "Three-Dimensional Natural Convection in a Box: A Numerical Study," *Journal of Fluid Mechanics*, Vol. 83, Part 1, pp. 1-31.

## DISTRIBUTION LIST

- |    |  |    |
|----|--|----|
| 1. | Naval Weapons Support Center<br>Code 6042<br>Attn: Mr. Tony Buechler<br>Crane, IN 47522  | 5  |
| 2. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, VA 22304-6145   | 2  |
| 3. | Library, Code 0142<br>Naval Postgraduate School<br>Monterey, CA 93943-5002   | 2  |
| 4. | Superintendent<br>Naval Postgraduate School<br>Attn: Professor Y. Joshi, Code ME/Ji<br>Department of Mechanical Engineering<br>Monterey, CA 93943-5004       | 10 |
| 5. | Superintendent<br>Naval Postgraduate School<br>Attn: Professor M. D. Kelleher, Code ME/Kk<br>Department of Mechanical Engineering<br>Monterey, CA 93943-5004 | 1  |
| 6. | Research Office<br>Naval Postgraduate School<br>Code 08<br>Monterey, CA 93943  | 1  |